

```
1      .TITLE  SHEP, 'APPLE DOS'
2      *      #5.1   6/2/78
3      *      8 BIT ASSEMBLER
4      *      M6502
5      *
6      *****
7      *      (C)  COPYRIGHT 1978  APPLE COMPUTER, INC
8      *****
9      *
10     1E00    ORG1    EQU    $1E00
11     2000    ORG2    EQU    $2000
12     3D00    DISKID  EQU    $3D00
13     3800    ASC1    EQU    $3800
14     3A8F    AEC1    EQU    $3A8F
15     3D00    ASC2    EQU    $3D00
16     3FFF    AEC2    EQU    $3FFF
17     4000    EDDS    EQU    ORG2+$2000
18
```

			PAGE	
			ORG	ORG1
19	0000			
20				
21		DOSREL		
22	1E00 AD4E1F		LDA	RSPAGE ; RESET START PAGE TO NORMAL
23	1E03 8D0B20		STA	ASTART+1
24				
25	1E06 A920		LDA	#DBINIT/256 ; RESET PI RTN TO NORMAL
26	1E08 8D5337		STA	DI3+2
27	1E0B A976		LDA	#DBINIT&255
28	1E0D 8D5237		STA	DI3+1
29				
30	1E10 4C161E		JMP	DRO
31				
32	1E13 4C0A21	RDONE	JMP	DBVECT+3
33				

PAGE

```

34      ;
35      ; GET RELOCATION PARMS
36      ;
37      ; DRO
38 1E16 A9BF      LDA    $$BF      ; START AT BFOO
39 1E18 8D4100    STA    ZPGWRK+1  ; TO LOOK FOR
40 1E1B A200      LDX    #0        ; HIGH RAM
41 1E1D 8E4000    STX    ZPGWRK
42 1E20 A140      LDA    (ZPGWRK,X) ; GET BYTE
43 1E22 49FF      EOR    $$FF      ; EX OR
44 1E24 8140      STA    (ZPGWRK,X) ; SET IT
45 1E26 C140      CMP    (ZPGWRK,X) ; DID IT TAKE
46 1E28 F005      BEQ    DR2        ; BR IF TOOK
47 1E2A CE4100    DEC    ZPGWRK+1  ; NOT RAM
48 1E2D D0F1      BNE    DR1        ; TRY NEXT PAGE
49      ;
50      ; DR2
51 1E2F 49FF      EOR    $$FF      ; RESTORE DATA
52 1E31 8140      STA    (ZPGWRK,X)
53      ;
54 1E33 AC4100    LDY    ZPGWRK+1
55 1E36 C8        INY
56 1E37 8C511F    STY    NEPAGE    ; NEW END OF DOS
57 1E3A 38        SEC
58 1E3B 98        TYA
59 1E3C ED521F    SBC    DOSLNG    ; MINUS DOS LENGTH
60 1E3F 8D501F    STA    NSPAGE    ; IS NEW START OF DOS
61 1E42 38        SEC
62 1E43 ED4E1F    SBC    RSPAGE    ; MINUS OLD DOS START
63 1E46 F0CB      BEQ    RDONE      ; (BRIF NO DELTA)
64 1E48 8D531F    STA    DELTA      ; IS DELTA
65

```

PAGE

66				
67				
68				
69				
70		DR3		
71	1E4B	BD0D1F	LDA	ADRTAB+1, X
72	1E4E	AB	TAY	
73	1E4F	BD0E1F	LDA	ADRTAB+2, X
74	1E52	8D4100	STA	ZPGWRK+1
75	1E55	4C661E	JMP	DR5
76				
77		DR4		
78	1E58	1B	CLC	
79	1E59	B140	LDA	(ZPGWRK), Y
80	1E5B	6D531F	ADC	DELTA
81	1E5E	9140	STA	(ZPGWRK), Y
82	1E60	CB	INY	
83	1E61	D003	BNE	DR5
84	1E63	EE4100	INC	ZPGWRK+1
85	1E66	CB	DR5	INY
86	1E67	D003	BNE	DR6
87	1E69	EE4100	INC	ZPGWRK+1
88				
89		DR6		
90	1E6C	AD4100	LDA	ZPGWRK+1
91	1E6F	DD101F	CMP	ADRTAB+4, X
92	1E72	90E4	BCC	DR4
93	1E74	9B	TYA	
94	1E75	DD0F1F	CMP	ADRTAB+3, X
95	1E78	90DE	BCC	DR4
96				
97	1E7A	8A	TXA	
98	1E7B	1B	CLC	
99	1E7C	6904	ADC	#4
100	1E7E	AA	TAX	
101	1E7F	EC0C1F	CPX	ADRTAB
102	1E82	90C7	BCC	DR3
103				

			PAGE	
104				
105			RELOCATE CODE	
106				
107	1E84 A200		LDX	#0
108	1E86 8E8033	DR7	STX	TEMP1
109				
110	1E89 BD361F		LDA	CDETAB+1, X ; GET A START OF CODE ADR
111	1E8C 8D4000		STA	ZPGWRK ; PUT IN ZPG
112	1E8F BD371F		LDA	CDETAB+2, X
113	1E92 8D4100		STA	ZPGWRK+1
114				
115	1E95 A200	DR8	LDX	#0
116	1E97 A140		LDA	(ZPGWRK, X) ; GET OP CODE
117	1E99 20BEF8		JSR	INSDS2 ; GO FIND OUT HOW LONG
118				
119	1E9C AC2F00		LDY	LENGTH ; GET HOW LONG
120	1E9F C002		CPY	#2 ; IF IT AIN'T
121	1EA1 D011		BNE	DR9 ; 3 THEN DON'T RELOC
122	1EA3 B140		LDA	(ZPGWRK), Y ; GET PAGE FROM INST
123	1EA5 CD4E1F		CMP	RSPAGE ; IF PAGE < REL START
124	1EA8 900A		BCC	DR9 ; THEN IGNOR
125	1EAA CD4F1F		CMP	REPAGE ; IF PAGE >= REL END
126	1EAD B005		BCS	DR9 ; THEN IGNORE
127	1EAF 6D531F		ADC	DELTA ; ELSE ADD DELTA
128	1EB2 9140		STA	(ZPGWRK), Y ; TO RELOCATE
129				
130	1EB4 38	DR9	SEC	
131	1EB5 AD2F00		LDA	LENGTH ; ADD LENGTH
132	1EB8 6D4000		ADC	ZPGWRK ; TO PC
133	1EBB 8D4000		STA	ZPGWRK
134	1EBE A900		LDA	#0
135	1EC0 6D4100		ADC	ZPGWRK+1
136	1EC3 8D4100		STA	ZPGWRK+1
137				
138	1EC6 AE8033		LDX	TEMP1 ; CHECK FOR END
139	1EC9 DD391F		CMP	CDETAB+4, X ; OF CODE SEGMENT
140	1ECC 90C7		BCC	DR8 ; BR NOT END
141	1ECE AD4000		LDA	ZPGWRK
142	1ED1 DD381F		CMP	CDETAB+3, X
143	1ED4 90BF		BCC	DR8 ; BR NOT END
144				
145	1ED6 8A		TXA	
146	1ED7 18		CLC	
147	1ED8 6904		ADC	#4 ; INCREMENT TABLE INDEX
148	1EDA AA		TAX	
149	1EDB EC351F		CPX	CDETAB ; DONE
150	1EDE 90A6		BCC	DR7 ; BR IF NOT
151				
152				

		PAGE		
153				
154			MOVE TO RELOCATED CODE	
155				
156	1EE0 AD4E1F	LDA	RSPAGE	
157	1EE3 8D4100	STA	ZPGWRK+1	; ZPGWRK=FROM
158	1EE6 AD501F	LDA	NSPAGE	
159	1EE9 8D4300	STA	ZPGFCB+1	; ZPGFCB = TOO
160	1EEC A900	LDA	#0	
161	1EEE 8D4000	STA	ZPGWRK	
162	1EF1 8D4200	STA	ZPGFCB	
163	1EF4 98	TYA		
164				
165	1EF5 B140	DR10	LDA (ZPGWRK),Y	; BYTE FROM
166	1EF7 9142		STA (ZPGFCB),Y	; BYTE TO
167	1EF9 C8		INY	; INCREMENT
168	1EFA D0F9		BNE DR10	; BR NOT FULL PAGE
169	1EFC CE541F		DEC DPGCNT	; DECREMENT PAGE CNT
170	1EFF F008		BEQ DR11	; BR IF DONE
171	1F01 EE4100		INC ZPGWRK+1	; INC FROM PAGE
172	1F04 EE4300		INC ZPGFCB+1	; INC TOO PAGE
173	1F07 D0EC		BNE DR10	; MOVE PAGE
174				
175	1F09 4C131E	DR11	JMP RDONE	; DONE
176				

			PAGE	
177	0040	DEPAGE	EQU	EDOS/256
178	0020	DSPAGE	EQU	START/256
179	F88E	INSDS2	EQU	\$F88E
180	002F	LENGTH	EQU	\$2F
181	1F0C 1C	ADRTAB	DB	7*4
182	1F0D 0020		DB	@@SAT1, @@EAT1
	1F0F 4E20			
183	1F11 5020		DB	@@RUN, @@RUN+2
	1F13 5220			
184	1F15 5A20		DB	@@IBVT+2, @@IBVT+4
	1F17 5C20			
185	1F19 7220		DB	@@AS2VT+6, @@AS2VT+8
	1F1B 7420			
186	1F1D 872B		DB	@@SAT2, @@EAT2
	1F1F C92B			
187	1F21 E437		DB	@@BAIOB, @@ADOSLD+2
	1F23 E837			
188	1F25 EE37		DB	@@IBDCTP, @@IBDCTP+2
	1F27 F037			
189	1F29 0000		DB	@0, @0
	1F2B 0000			
190	1F2D 0000		DB	@0, @0
	1F2F 0000			
191	1F31 0000		DB	@0, @0
	1F33 0000			
192		CDETAB		
193	1F35 14		DB	5*4
194	1F36 7620		DB	@@SC1, @@EC1
	1F38 8F29			
195	1F3A C92B		DB	@@SC2, @@EC2
	1F3C 7633			
196	1F3E 0037		DB	@@SC3, @@EC3
	1F40 E037			
197	1F42 0038		DB	@@ASC1, @@AEC1
	1F44 8F3A			
198	1F46 003D		DB	@@ASC2, @@AEC2
	1F48 FF3F			
199	1F4A 0000		DB	@0, @0
	1F4C 0000			
200				
201	1F4E 20	RSPAGE	DB	DSPAGE
202	1F4F 40	REPAGE	DB	DEPAGE
203				
204	1F50 00	NSPAGE	DB	0
205	1F51 00	NEPAGE	DB	0
206				
207	1F52 20	DOSLNG	DB	DEPAGE-DSPAGE
208				
209	1F53 00	DELTA	DB	0
210	1F54 20	DPGCNT	DB	DEPAGE-DSPAGE
211				

		PAGE			
		ORG		ORG2	
212	1F55				
213					
214					
215					
216			START		
217			SAT1		
218	2000 D31F		FTAB	DB	@@*-45 ; START OF FTABS
219	2002 2121		CINA	DB	@@CHRIN ; CHAR IN ADR
220	2004 4721		COUTA	DB	@@CHROUT ; CHAR OUT ADR
221	2006 3B2B		FN1ADR	DB	@@FNAME1
222	2008 5B2B		FN2ADR	DB	@@FNAME2
223	200A 001E		ASTART	DB	@@DOSREL ; CHANGED TO START BY RELOCATE
224	200C 8A35		CCBADR	DB	@@CCB
225					
226			OUTSVT		; CHAR OUTPUT STATE VECTOR TABLE
227	200E 6A21			DB	@@COS0-1
228	2010 8921			DB	@@COS1-1
229	2012 9A21			DB	@@COS2-1
230	2014 A621			DB	@@COS3-1
231	2016 BC21			DB	@@COS4-1
232	2018 CB21			DB	@@COS5-1
233	201A D721			DB	@@COS6-1
234					
235					
236					
237					
238	201C 6727			DB	@@EINIT-1
239	201E 0026			DB	@@ELOAD-1
240	2020 8E25			DB	@@ESAVE-1
241	2022 9126			DB	@@ERUN-1
242	2024 A726			DB	@@ECHAIN-1
243	2026 7724			DB	@@EDEL-1
244	2028 8524			DB	@@ELOCK-1
245	202A 8924			DB	@@EUNLK-1
246	202C F824			DB	@@ECLOSE-1
247	202E BD26			DB	@@EREAD-1
248	2030 F126			DB	@@EEXEC-1
249	2032 B226			DB	@@EWRITE-1
250	2034 0527			DB	@@EPOS-1
251	2036 B524			DB	@@EOPEN-1
252	2038 AA24			DB	@@EAPND-1
253	203A 9224			DB	@@EREN-1
254	203C 3627			DB	@@ECAT-1
255	203E 4624			DB	@@EMON-1
256	2040 5024			DB	@@ENOMON-1
257	2042 2024			DB	@@EPR-1
258	2044 3324			DB	@@EIN-1
259	2046 6424			DB	@@EMAXF-1
260	2048 4427			DB	@@EAS-1
261	204A 4025			DB	@@EBSV-1
262	204C 6C25			DB	@@EBLD-1
263					
264			EAT1		

PAGE

```

265
266
267
268
269 204E 36E8 CHAIN DB @@IBCHN
270 2050 9D26 RUN DB @@IBRUN
271 2052 E3E3 BREAK DB @@IBBRK
272 2054 00E0 GO DB @@IBGO
273 2056 03E0 CONT DB @@IBCONT ; BASIC CONT ENTRY POINT
274 2058 36E8 IBVT DB @@IBCHN, @@IBRUN, @@IBBRK
    205A 9D26
    205C E3E3
275 205E 00E0 DB @@IBGO, @@IBCONT
    2060 03E0
276 000A IBVTL EQU *-IBVT
277
278 2062 D2D7 AS1VT DB @@ASRUN1, @@ASRUN1, @@ASBRK1
    2064 D2D7
    2066 65D8
279 2068 00E0 DB @@IBGO, @@0
    206A 0000
280 000A AS1VTL EQU *-AS1VT
281
282 206C D40F AS2VT DB @@ASRUN2, @@ASRUN2, @@ASBRK2
    206E D40F
    2070 6710
283 2072 7620 DB @@DBINIT, @@0
    2074 0000
284 000A AS2VTL EQU *-AS2VT
285

```

PAGE

```

286      ;
287      ; EQUATES REQD TO FIND THINGS IN APPLE II
288      ;
289      FE93      SETVID EQU      $FE93
290      FE89      SETKBD EQU      $FE89
291      0033      PROMPT EQU      $33          ; PROMPT CHAR
292      0036      OUTSW EQU      $36          ; OUTPUT VECTOR SWITCH
293      0038      INSW EQU      $38          ; INPUT VECTOR SWITCH
294      0040      ZPGWRK EQU      $40          ; ZERO PAGE WORK CELL
295      0044      CNUM EQU      $44          ; CONVERTED NUMERIC
296      0200      LBUFF EQU      $200        ; LINE BUFFER
297      FB63      MULT EQU      $FB63        ; MULT ROUTINE
298      FE8B      INPRT EQU      $FE8B        ; SET IN PORT
299      FE95      OUTPRT EQU      $FE95        ; SET OUT PORT
300      E836      IBCHN EQU      $E836        ; BASIC RUN
301      004A      IBLMEM EQU      $4A          ; BASIC LOW MEM
302      004C      IBHMEM EQU      $4C          ; INTEGER BASIC HIMEM
303      00CA      IBSOP EQU      $CA          ; INTEGER BASIC START OF CGM
304      E3E3      IBBRK EQU      $E3E3        ; BASIC BREAK
305      E000      IBGD EQU      $E000        ; BASIC ENTRY POINT
306      E003      IBCONT EQU      $E003        ; BASIC CONTINUE ENTRY POINT
307      00CC      IBSDV EQU      $CC          ; BASIC START OF VARIABLES
308      0067      ASSOP EQU      $67          ; AS START OF PROGRAM
309      00AF      ASEOP EQU      $AF          ; AS END OF PROGRAM
310      0069      ASEDP2 EQU      $69          ; AS END-OF PGM 2
311      0073      ASHM1 EQU      $73          ; AS HIGH MEM 1
312      006F      ASHM2 EQU      $6F          ; AS HIGH MEM 2
313      0067      ASLMEM EQU      ASSOP        ; AS LOW MEM
314      D7D2      ASRUN1 EQU      $D7D2        ; AS ROM RUN
315      OFD4      ASRUN2 EQU      $OFD4        ; AS RAM CLEAR
316      D865      ASBRK1 EQU      $D865        ; AS ROM BREAK
317      1067      ASBRK2 EQU      $1067        ; AS RAM BREAK
318      E000      AITSTL EQU      $E000        ; AS 1 IB TEST LOC
319      004C      ATSTV EQU      $4C          ; AS TEST VALUE
320      0020      ITSTV EQU      $20          ; IB TEST VALUE
321      002E      BOOTSL EQU      $2E          ; BOOT FROM SLOT
322      0042      ZPGFCB EQU      $42          ; ZERO PAGE WORK CELL
323      FC58      HOME EQU      $FC58
324      FDED      PRINT EQU      $FDED
325      FDOC      GETKEY EQU      $FDOC
326

```


PAGE

```

327 ;
328 ; DOS BASIC INTERPRETER - INITIAL ENTRY
329 ;
330 SC1
331 DBINIT
332 2076 AD00E0 LDA AITSTL ; GET APPLESOFT/IB TEST
333 2079 4920 EOR #ITSTV ; IF AS THEN
334 207B D011 BNE IAS1 ; GO TO AS INIT
335 ; ; ELSE INIT FOR IB
336 207D 8D802B STA ASIBSW ; SET SW FOR IB
337 2080 A20A LDX #IBVTL ; GET IB VT LENGTH
338 2082 BD5720 IIB1 LDA IBVT-1,X ; MOVE IB ADDR
339 2085 9D4D20 STA IBASVT-1,X
340 2088 CA DEX
341 2089 D0F7 BNE IIB1
342 208B 4C9E20 JMP INITAA
343 ;
344 IAS1
345 208E A940 LDA #$40 ; INDICATE ROM APPLESOFT
346 2090 8D802B STA ASIBSW
347 2093 A20A LDX #AS1VTL
348 2095 BD6120 IAS1A LDA AS1VT-1,X ; MOVE ROM AS ADRS
349 2098 9D4D20 STA IBASVT-1,X
350 209B CA DEX
351 209C D0F7 BNE IAS1A
352 ;
353 INITAA
354 209E 38 SEC ; INDICATE INIT
355 209F B001 BCS INITA
356 DBRST
357 20A1 18 CLC ; INDICATE RESET
358 ;
359 INITA
360 20A2 08 PHP ; SAVE INIT/RESET
361 20A3 2093FE JSR SETVID
362 20A6 2089FE JSR SETKBD
363 20A9 206529 JSR MVISW ; SET INSW
364 20AC 207A29 JSR MVOSW ; SET OUTSW
365 20AF A970 LDA #MC+MI+MO ; SET MONITOR MODES
366 20B1 8D272B STA MONMOD
367 ;
368 20B4 A900 LDA #0
369 20B6 8D1B2B STA OSTATE ; CLEAR OUTSTATE AND EXECUTE STATE
370 20B9 8D7D2B STA ESTATE ; EXECUTE STATE
371 20BC 28 PLP ; GET INIT/RESET
372 20BD 6A RORA ; SHIFT CARRY TO MSB
373 20BE 8D1A2B STA ISTATE ; SAVE INSTATE
374 20C1 3003 BMI INITB ; BR IF INIT
375 20C3 6C5620 JMP (CONT) ; GO TO CONTINUE ENTRY
376 20C6 6C5420 INITB JMP (GO) ; GO TO GO ENTRY
377

```


			PAGE		
378		INITC			
379	20C9 205E21		JSR	SVRGSA	; GO SAVE OTHER REGS
380	20CC 0A		ASLA		; OF ISTATE NOT ON
381	20CD 100E		BPL	INITD	; THEN NOT RAM AS
382	20CF 8D802B		STA	ASIBSW	; SET RAM AS
383	20D2 A20A		LDX	#AS2VTL	
384	20D4 BD6B20	IAS2A	LDA	AS2VT-1, X	; MOVE RAM AS ADRS
385	20D7 9D4D20		STA	IBASVT-1, X	
386	20DA CA		DEX		
387	20DB D0F7		BNE	IAS2A	
388					
389		INITD			
390	20DD AD7B2B		LDA	DFNFTS	; GO BUILD FILE TABS
391	20E0 8D202B		STA	CNFTBS	; AND SET MEM BOUNDS
392	20E3 20E728		JSR	BLDFTB	
393	20E6 207128		JSR	CLRSTS	; SET IN AND OUT STATES TO ZERO
394	20E9 A21D		LDX	#IFBL	
395	20EB BD0721	INITE	LDA	DBVECT, X	; MOVE RESTART VECTORS
396	20EE 9D8003		STA	#380, X	
397	20F1 CA		DEX		
398	20F2 10F7		BPL	INITE	
399	20F4 AD282B		LDA	CMDNO	; IF NOT BOOT
400	20F7 D00A		BNE	INITF	; THEN DONE
401	20F9 AD3B2B		LDA	FNAME1	; IF FN1
402	20FC 49A0		EOR	#A0	; NOT GIVEN
403	20FE F003		BEG	INITF	; THEN DONE
404	2100 4C9226		JMP	ERUN	; ELSE RUN
405					
406		IFB			
407	2103 38	INITF	SEC		
408	2104 4C0022		JMP	ORTN	
409					
410	2107 4CA120	DBVECT	JMP	DWRST	
411	210A 4C7620		JMP	DBINIT	
412	210D 4CC92B		JMP	DOSENT	
413	2110 4C003D		JMP	DISKIO	
414		CCBLDR			
415	2113 AD0D20		LDA	CCBADR+1	
416	2116 AC0C20		LDY	CCBADR	
417	2119 60		RTS		
418		IOBLDR			
419	211A AD882B		LDA	AIOB+1	
420	211D AC872B		LDY	AIOB	
421	2120 60		RTS		
422	001D	IFBL	EQU	*-IFB-1	
423					

PAGE

```

424      ;
425      ;   CHRIN - CHAR RCVD VIA IN SWITCH
426      ;
427      CHRIN
428 2121 8D252B      STA      SVA
429 2124 AD1A2B      LDA      ISTATE      ; IF NOT DISKIN
430 2127 F00B        BEQ      CHIN1      ; THEN BRANCH, ELSE
431 2129 309E        BMI      INITC
432 212B 205B21      JSR      SVREGS      ; SAVE REGS
433 212E 4C8A27      JMP      ICFD        ; AND GET CHAR FROM DISK
434      CHIN1
435 2131 AD7D2B      LDA      ESTATE
436 2134 F006        BEQ      CHIN2
437 2136 205E21      JSR      SVRGSA      ; SAVE REGS
438 2139 4CAA27      JMP      NXTEXC
439      CHIN2
440 213C A903        LDA      #3          ; SET OUT CHAR
441 213E 8D1B2B      STA      OSTATE      ; STATE TO INPUT ECHO
442 2141 AD252B      LDA      SVA
443 2144 6C1E2B      JMP      (SVINS)      ; CONTINUE WITH CHAR PROCESS
444      ;
445      ;   CHROUT - CHAR RCVD VIA OUTPUT SWITCH
446      ;
447      CHROUT
448 2147 205B21      JSR      SVREGS      ; SAVE REGS
449      ;
450 214A AD1B2B      LDA      OSTATE      ; GET OUT SPARE
451 214D 0A          ASLA
452 214E AA          TAX
453 214F BD0F20      LDA      OUTSVT+1,X  ; GET ROUTINE ADR
454 2152 4B          PHA
455 2153 BD0E20      LDA      OUTSVT,X
456 2156 4B          PHA
457 2157 AD252B      LDA      SVA
458 215A 60          RTS                ; GO TO ROUTINE
459      ;
460      ;   SVREGS - SAVE REGS WHILE PROCESSING CHARS
461      ;
462      SVREGS
463 215B 8D252B      STA      SVA          ; SAVE ACU
464      SVRGSA
465 215E 8E232B      STX      SVX          ; SAVE X
466 2161 8C242B      STY      SVY          ; SAVE Y
467 2164 BA          TSX                ; SAVE STACK
468 2165 E8          INX
469 2166 E8          INX
470 2167 8E222B      STX      SVSTK
471 216A 60          RTS                ; DONE
472

```



```

PAGE
473 ;
474 ; COS0 - 1ST CHAR OF PRINTED OUTPUT LINE
475 ; CHECK FOR CNTL-D
476 ;
477 COS0
478 216B AE1A2B LDX ISTATE ; IS IN STATE NOT ZERO
479 216E F00B BEQ COS01
480 2170 C9BF CMP #'?'+$80 ; THEN IS THIS ?
481 2172 F064 BEQ COS6 ; THEN PRINT ONLY IF MONITOR
482 2174 C533 CMP PROMPT
483 2176 F060 BEQ COS6
484 COS01
485 2178 A202 LDX #2
486 217A 8E1B2B STX OSTATE
487 217D CD7C2B CMP CCHAR ; IF NOT CNTL-D
488 2180 D019 BNE COS2 ; THEN GO TO STATE 2
489 2182 CA DEX
490 2183 8E1B2B STX OSTATE ; ELSE STATE = 1
491 2186 CA DEX
492 2187 8E262B STX LBUFD ; AND LBUFD=0
493 ;
494 ; COS1 - ACCUMULATE CMD FROM PRINTED OUTPUT
495 ;
496 COS1
497 218A AE262B LDX LBUFD ; GET LINE BUFF DISPL
498 218D 9D0002 COS1A STA LBUFD,X ; PUT CHAR IN BUFF
499 2190 E8 INX ; INCR PTR
500 2191 8E262B STX LBUFD ; SAVE PTR
501 2194 C98D CMP #$8D ; WAS THIS A CR
502 2196 D057 BNE CMDRTN ; IF NOT THEN PR CHAR
503 ;
504 2198 4C1E22 JMP SCNCMD ; GO SCAN COMMAND
505 ;
506 ; COS2 - PRINTED OUTPUT, NOT FIRST CHAR
507 ;
508 COS2
509 219B C98D CMP #$8D ; IS IT A CR
510 219D D040 BNE PRRTN ; BR IF NOT
511 219F A200 LDX #0 ; SET FOR POSSIBLE C-D NEXT
512 21A1 8E1B2B STX OSTATE ; NEXT STATE
513 21A4 4CDF21 JMP PRRTN ; GO PRINT CHAR
514

```


PAGE

```

515 ;
516 ; COS3 - KEY IN ECHO PRINT
517 ;
518 COS3
519 21A7 A200 LDX #0
520 21A9 8E1B2B STX OSTATE ; RESET OUT STATE
521 21AC C98D CMP #$8D ; IS IT CR
522 21AE F007 BEQ COS3A ; IF CR THEN CMD CHECK
523 21B0 AD7D2B LDA ESTATE ; ELSE: IF NOT EXECUTE
524 21B3 F02A BEQ PRRTN ; THEN PRINT CHAR
525 21B5 D040 BNE DRTNI ; ELSE: PRINT IF MON INPUT
526 COS3A
527 ;
528 21B7 AE232B LDX SVX ; GET LINE INDEX
529 21BA 4C8D21 JMP COS1A
530 ;
531 ; COS4 - DISK OUTPUT MODE
532 ;
533 COS4
534 21BD C98D CMP #$8D ; IS IT CR
535 21BF D005 BNE COS4A ; BR IF NOT CR
536 21C1 A905 LDA #5 ; SET STATE FOR CNTL-D
537 21C3 8D1B2B STA OSTATE ; EXAMINE
538 21C6 207627 COS4A JSR OCTD ; GO OUTPUT CJHAR TO DISK
539 21C9 4CF321 JMP DRTNO ; GO TO DATA RETURN (OUT)
540 ;
541 ; COS5 - DISK OUTPUT MODE - 1ST CHAR OF A LINE
542 ;
543 COS5
544 21CC CD7C2B CMP CCHAR ; IS IT CNTL D
545 21CF F09A BEQ COS0 ; BR IF CNTL- D
546 21D1 A204 LDX #4
547 21D3 8E1B2B STX OSTATE ; SET NEW OUT STATE
548 21D6 D0E5 BNE COS4 ; BR IF NOT CNTL D
549 ;
550 ; COS6 - DISK INPUT ECHO
551 ;
552 21D8 A900 COS6 LDA #0
553 21DA 8D1B2B STA OSTATE ; RESET OUT STATE = 0
554 21DD F018 BEQ DRTNI ; GO TO DATA IN RETURN
555

```



```

PAGE
556
557
558
559 21DF 18      ;
560 21E0 901E    PRRTN  CLC          ; INDICATE PRINT
561              BCC      ORTN      ; GO RETURN
562
563              ;
564              ; CMDRTN - PRINT CHAR IF MONITOR CMBS MODE
565              ; DRTNO - PRINT CHAR IF MONITOR DATA OUT
566              ; DRTNI - PRINT CHAR IF MONITOR DATA IN
567
568              ;
569              ; CERTN
570 21E2 AD0002    LDA      LBUF      ; CHECK FOR PRINTED COMMAND
571 21E5 CD7C2B    CMP      CCHAR
572 21E8 F005      BEQ      CMDRTN    ; IF PC THEN NO RESET X REG
573 21EA A200      LDX      #0        ; RESET TO SOL
574 21EC 8E232B    STX      SVX
575
576 21EF A940      CMDRTN  LDA      #MC
577 21F1 D006      BNE      MODECK
578 21F3 A910      DRTNO   LDA      #MO
579 21F5 D002      BNE      MODECK
580 21F7 A920      DRTNI   LDA      #MI
581
582              ;
583 21F9 18        MODECK  CLC          ; ASSUME PRINT
584 21FA 2D272B    AND      MONMOD    ; AND WITH MODE
585 21FD D001      BNE      ORTN      ; BR IF PRINT
586 21FF 3B        SEC
587
588              ;
589 2200 AE222B    ORTN    LDX      SVSTK ; GET STACK
590 2203 9A        LDREGS TXS          ; RESTORE STACK
591
592 2204 AD252B    LDA      SVA        ; ACU
593 2207 AC242B    LDY      SVY        ; Y
594 220A AE232B    LDX      SVX        ; X
595 220D 9001      BCC      ORTN1     ; BR IF PRINT
596 220F 60        RTS              ; BY PASS PRINT
597
598 2210 6C1C2B    ORTN1   JMP      (SVOUTS)
599
600              ;
601              ; PRCRIF - PRINT CR IF MON CMDS
602              ;
603              ;
604              ; PRCRIF
605 2213 2C272B    BIT      MONMOD    ; IF NOT MON CMDS
606 2216 5005      BVC      PRCIFR    ; THEN RETURN
607 2218 A98D      LDA      #8D      ; ELSE PRINT CR
608 221A 201022    JSR      ORTN1
609 221D 60        PRCIFR  RTS
610
611

```


		PAGE		
603				
604			SCNCMD - SCAN A COMMAND	
605				
606			SCNCMD	
607	221E	A0FF	LDY	##FF
608	2220	8C282B	STY	CMDNO
609	2223	CB	INY	
				; RESET COMMAND NUMBER
				; INCR TABLE INDEX
610			SCO	
611	2224	EE282B	INC	CMDNO
				; INCR CMD NO
612	2227	A200	LDX	#0
				; RESET LINE INDEX TO 0
613	2229	0B	PHP	
				; SAVE EQ STATUS
614	222A	BD0002	LDA	LBUFF, X
				; GET 1ST LINE CHAR
615	222D	CD7C2B	CMP	CCHAR
				; IS IT CONTROL D
616	2230	D001	BNE	SCOA
				; BR /IF NOT
617	2232	EB	INX	
				; INCR OVER CNTLD
618	2233	BE262B	STX	LBUFD
619				
620			SC1X	
621	2236	20CC23	JSR	GNBC
				; GET NON BLANK INPUT CHAR
622	2239	297F	AND	##7F
				; MSB OF CHAR OFF
623	223B	598F29	EOR	CMDNTB, Y
				; EOR WITH INPUT
624	223E	CB	INY	
				; INCREMENT TABLE INDEX
625	223F	0A	ASLA	
				; IF MSB OF EOR RESULT ON
626	2240	F002	BEQ	SC1A
				; IF RESULT NOT NOW ZERO
627	2242	6B	PLA	
				; THEN INPUT DOES NOT
628	2243	0B	PHP	
				; EQUAL ENTRY
629	2244	90F0	BCC	SC1X
				; LOOP FOR END OF ENTRY
630				
631	2246	2B	PLP	
				; IF INPUT EQUALS END
632	2247	F01E	BEQ	SYNTAX
				; THEN GO SYNTAX
633				
634	2249	B98F29	LDA	CMDNTB, Y
				; IF NEXT TABLE CHAR NOT ZERO
635	224C	D0D6	BNE	SCO
				; THEN SCAN THE NEXT TABLE ENTRY
636				; COMMAND NOT FOUND
637	224E	AD0002	LDA	LBUFF
				; LINE IS A CNOTROL-D
638	2251	CD7C2B	CMP	CCHAR
				; THEN THIS IS A
639	2254	F003	BEQ	CNF1
				; POSSIBLE SYNTAX ERROR, ELSE
640	2256	4CDF21	JMP	PRRTN
				; ITS A BASIC INPUT LINE
641			CNF1	
642	2259	20CC23	JSR	GNBC
				; GET NON BLANK CHAR
643	225C	D006	BNE	CSERR
				; BR IF CR
644	225E	20712B	JSR	CLRSTS
				; CLEAR THE STATES
645	2261	4CEF21	JMP	CMDRTN
				; CNTL-D ONLY
646				
647	2264	4CE227	JMP	ESYNTX
648				

PAGE

```

649      ;
650      ; SYNTAX - FIGURE OUT WHAT WE GOT HERE
651      ;
652      SYNTAX
653 2267 AD282B LDA CMDNO ; CMDNO=CMDNO*2
654 226A 0A ASLA
655 226B 8D282B STA CMDNO
656      ;
657 226E A8 TAY
658 226F A920 LDA #FN1
659 2271 390E2A AND CMDSTB,Y ; IS FN1 REQD
660 2274 F05C BEQ SN10 ; BR IF NOT
661 2276 20C522 JSR CLRFS
662 2279 08 PHP ; SAVE EQ STATUS
663      ;
664      SN2
665 227A 20CC23 JSR GNBC ; GET NON BLANK CHAR
666 227D F014 BEQ SN6 ; BR IF CR OR COMMA
667 227F D005 BNE SN4 ; BR IF REAL CHAR
668 2281 20BB23 SN3 JSR GNXTC ; GO GET NEXT CHAR
669 2284 F00D BEQ SN6 ; BR IF COMMA OR CR
670 2286 993B2B SN4 STA FNAME1,Y ; PUT INTO FILENAME
671 2289 C8 INY ; INC FN INDEX
672 228A C040 CPY #64 ; ATFN CHAR LIMIT
673 228C 90F3 BCC SN3 ; BR IF NOT
674 228E 20BB23 SN5 JSR GNXTC ; LOOP UNTIL CR OR COMMA
675 2291 F0FB BEQ SN5
676      ;
677 2293 28 SN6 PLP ; WAS THIS FN2 L 00
678 2294 D00F BNE SN7 ; BR IF IT WAS
679      ;
680 2296 AC282B LDY CMDNO
681 2299 A910 LDA #FN2
682 229B 390E2A AND CMDSTB,Y ; IF FN2 NOT REQD THEN
683 229E F00C BEQ SN8 ; BRANCH
684      ;
685 22A0 A020 LDY #32 ; SET FN2 INDEX
686 22A2 08 PHP ; INDICATE FN2 SEEK
687 22A3 D0D5 BNE SN2 ; GO LOOK FOR FN2
688      ;
689 22A5 AD5B2B SN7 LDA FNAME2 ; IF 1ST CHAR OF
690 22A8 C9A0 CMP #A0 ; FN2 IS BLANK THEN
691 22AA F016 BEQ SERR1 ; SYNTAX ERROR
692      ;
693 22AC AD3B2B SN8 LDA FNAME1 ; IF 1ST CHAR OF
694 22AF C9A0 CMP #A0 ; FN1 IS NOT BLANK
695 22B1 D048 BNE SOPTS ; THEN GO LOOK FOR OPTIONS
696      ;
697 22B3 AC282B LDY CMDNO
698 22B6 A9C0 LDA #NPB+NPE ; IF CMD MUST HAVE FILENAME
699 22B8 390E2A AND CMDSTB,Y ; THEN
700 22BB F005 BEQ SERR1 ; THIS IS ERROR, ELSE
701      ;
702 22BD 103C BPL SOPTS ; ITS EXECUTABLE WITHOUT

```


703	22BF	4C4E22		JMP	CNF	; OR ITS PLAIN OLD LOAD, SAVE, RUN
704			,			
705	22C2	4CE227	SERR1	JMP	ESYNTAX	
706			,			
707			CLRFNS			
708	22C5	A900		LDA	#0	
709	22C7	A040		LDY	#64	
710	22C9	A9A0		LDA	#\$A0	
711	22CB	993A2B	SN1	STA	FNAME1-1,Y	; CLEAR FN1, FN2
712	22CE	88		DEY		
713	22CF	D0FA		BNE	SN1	
714	22D1	60		RTS		
715						

		PAGE	
716		SN10	; FILE NAMES NOT REQD
717	22D2 8D3B2B	STA	FNAME1
718	22D5 A90C	LDA	#NUM1+NUM2 ; IF NEITHER NUM1
719	22D7 390E2A	AND	CMDSTB,Y ; OR NUM2 IS REQD
720	22DA F01F	BEQ	SOPTS ; THEN GO LOOK AT OPTIONS
721			
722	22DC 20E123	JSR	GETNUM ; GO GET NUMERICS
723	22DF B0E1	BCS	SERR1
724			
725	22E1 A8	TAY	; IF HIGH DIGIT NOT
726	22E2 D0DE	BNE	SERR1 ; ZERO THEN BAD
727			
728	22E4 E011	CPX	#17 ; IF LOW DIGIT GT 16
729	22E6 B0DA	BCS	SERR1 ; THEN BAD
730			
731	22E8 AC282B	LDY	CMDND
732	22EB A908	LDA	#NUM1
733	22ED 390E2A	AND	CMDSTB,Y ; IF WE WANT NUM2
734	22F0 F006	BEQ	SN11
735			
736	22F2 E008	CPX	#8 ; IF NUM2>1
737	22F4 B0CC	BCS	SERR1 ; THEN ERROR, ELSE
738	22F6 9003	BCC	SOPTS ; GO SCAN OPTIONS
739			
740		SN11	
741	22F8 BA	TXA	; IF NUM1=0
742	22F9 F0C7	BEQ	SERR1 ; THEN ERROR, ELSE
743			
744			

PAGE

```

745 ;
746 ; SOPTS - LOOK FOR SYNTAX OPTIONS
747 ;
748 ; SOPTS
749 22FB A900 LDA #0
750 22FD 8D2B2B STA INOPTS ; CLEAR INPUT OPTIONS
751 2300 8D3A2B STA IMBITS
752 2303 8D322B STA CL
753 2306 8D332B STA CL+1
754 2309 AD262B LDA LBUFD ; SET PASS 1
755 230C 8D292B STA TEMP1A
756 ;
757 230F 20CC23 SP1 JSR GNBC ; GO GET NON-BLANK CHAR
758 2312 D01F BNE SP2 ; BR IF NOT COMMA OR CR
759 2314 C98D CMP #8D ; IF CHAR IS COMMA
760 2316 D0F7 BNE SP1 ; THEN GO GET CHAR
761 ;
762 2318 AE282B LDY CMDNO ; OPTIONS INPUT = I
763 231B AD2B2B LDA INOPTS ; ALLOW OPTS = A
764 231E 1D0F2A ORA CMDSTB+1,X ; IF (A OR I)
765 2321 5D0F2A EOR CMDSTB+1,X XOR A NOT = 0 THEN
766 2324 D09C BNE SERR1 ; WE HAVE UNALLOWED OPTIONS
767 ;
768 2326 AE292B LDY TEMP1A ; IF THIS IS PASS 2
769 2329 F077 BEQ CMDGO ; THEN DONE,
770 232B 8D292B STA TEMP1A ; ELSE SET PASS
771 232E 8E262B STX LBUFD ; RESTORE LBUFD AND
772 2331 D0DC BNE SP1 ; GO DO PASS 2
773 ;
774 2333 A20A SP2 LDY #OPT1L ; COMPARE CHAR HAVE WITH
775 2335 DD3F2A SP3 CMP OPTAB1-1,X ; CHARS IN OPT TABLE
776 2338 F005 BEQ SP4 ; IF FOUND CONTINUE,
777 233A CA DEX
778 233B D0FB BNE SP3 ; IF NOT FOUND
779 233D F060 BEQ SERR2 ; THEN SYNTAX ERROR
780 ;
781 233F BD492A SP4 LDA OPTAB2-1,X ; IF CORRESPONDING OP TAB 2 IS
782 2342 3047 BMI SP8 ; MINUS THEN IT MONITOR BITS
783 2344 0D2B2B ORA INOPTS
784 2347 8D2B2B STA INOPTS
785 234A CA DEX
786 ;
787 234B 8E2A2B STX TEMP2A ; ELSE A NUMERIC MUST FOLLOW
788 234E 20E123 JSR GETNUM ; FOLLOW
789 2351 B04C BCS SERR2
790 ;
791 2353 AD2A2B LDA TEMP2A ; GET IOTION NUMBER
792 2356 0A ASLA ; MULT BY 4
793 2357 0A ASLA
794 2358 AB TAY
795 ;
796 2359 A545 LDA CNUM+1 ; IF RESULT NUM HI IS
797 235B D009 BNE SP5 ; GT 0, THEN GT LOW RANGE
798 235D A544 LDA CNUM ; TEST RESULT LOW

```



```

799 235F D9542A      CMP      OPTAB3,Y      ; WITH LOW RANGE (LOW)
800 2362 903B        BCC      SERR2          ; BR IF RESULT < LR
801 2364 A545         LDA      CNUM+1
802 2366 D9572A      SP5    CMP      OPTAB3+3,Y
803 2369 900B        BCC      SP6            ; BR IF LESS
804 236B D032         BNE      SERR2          ; BR IF GRGREATER
805 236D A544         LDA      CNUM
806 236F D9562A      CMP      OPTAB3+2,Y
807 2372 9002        BCC      SP6            ; BR IF LESS
808 2374 D029        BNE      SERR2          ; BR IF GREATER
809
810 2376 AD292B      SP6    LDA      TEMP1A      ; IF PASS 1, THEN
811 2379 D094        BNE      SP1            ; DONT STORE RESULT
812 237B 98          TYA
813 237C 4A          LSRA
814 237D AB          TAY
815
816 237E A545         LDA      CNUM+1      ; STORE THE RESULT
817 2380 992D2B      STA      CUROPT+1,Y
818 2383 A544         LDA      CNUM
819 2385 992C2B      STA      CUROPT,Y
820 2388 4C0F23      SP7    JMP      SP1      ; GO FOR NEXT OPT
821
822                SP8    ; MONITOR REQ
823 238B 48          PHA      ; SAVE TYPE REQ
824 238C A980        LDA      #CIO          ; SET OPTION OF CIO
825 238E 0D2B2B      ORA      INOPTS
826 2391 8D2B2B      STA      INOPTS
827 2394 68          PLA      ; RESTORE REQ
828 2395 297F        AND      #$7F          ; CLEAR CIO
829 2397 0D3A2B      ORA      IMBITS        ; OR WITH PREV IMBITS
830 239A 8D3A2B      STA      IMBITS
831 239D D0E9        BNE      SP7            ; GO FOR NEXT
832
833 239F 4CE227      SERR2  JMP      ESYNTAX
834

```


		PAGE		
835				
836				
837				
838				
839	23A2 20712B	JSR	CLRSTS	
840	23A5 20D623	JSR	CLRCCB	; GO CLEAR CCB
841	23A8 20AE23	JSR	ECMD	; GO EXECUTE
842	23AB 4CE221	JMP	CERTN	
843				
844	23AE AD282B	LDA	CMDNO	; COMMAND NO
845	23B1 AA	TAX		; IS CMD EXEC TAB INDEX
846	23B2 BD1D20	LDA	CMDETB+1, X	; GET CMD ADR
847	23B5 4B	PHA		; ONTO STACK
848	23B6 BD1C20	LDA	CMDETB, X	
849	23B9 4B	PHA		
850	23BA 60	RTS		; AND GOTO COMMAND
851				
852				
853				
854				
855	23BB AE262B	LDX	LBUFD	
856	23BE BD0002	LDA	LBUFF, X	; GET NEXT CHAR AND IF
857	23C1 C98D	CMP	##8D	; IT IS A CR
858	23C3 F006	BEG	GNXTCR	; THEN RETURN WITHOUT
859	23C5 EB	INX		; INCR TO NEXT CHAR
860	23C6 8E262B	STX	LBUFD	
861	23C9 C9AC	CMP	#, +\$80	; TEST FOR COMMA
862	23CB 60	GNXTCR RTS		
863				
864				
865				
866				
867	23CC 20BB23	JSR	GNXTC	; GO GET NEXT CHAR
868	23CF FOFA	BEG	GNXTCR	; BR IF COMMA OR CR
869	23D1 C9A0	CMP	##A0	; IS IT BLANK
870	23D3 F0F7	BEG	GNBC	; BR IF BLANK
871	23D5 60	RTS		; DONE
872				
873				
874				
875				
876	23D6 A900	LDA	#0	
877	23D8 A016	LDY	#CCBLEN	; CCBLENGTH
878	23DA 998935	STA	CCB-1, Y	; CLEAR BYTE
879	23DD 8B	DEY		
880	23DE D0FA	BNE	CLC1	
881	23E0 60	RTS		
882				


```

PAGE
883
884
885
886 GETNUM
887 23E1 A900 LDA #0 ; CLEAR WORK AREA
888 23E3 8544 STA CNUM
889 23E5 8545 STA CNUM+1
890
891 23E7 20CC23 GN2 JSR GNBC ; GET NEXT NON BLANK
892 23EA D006 BNE GN3 ; BR NOT COMMA OR CR
893 23EC A644 LDX CNUM ; X=RESULT LOW
894 23EE A545 LDA CNUM+1 ; Y=RESULT HI
895 23F0 18 CLC
896 23F1 60 RTS ; DONE
897
898 23F2 38 GN3 SEC
899 23F3 E9B0 SBC #$B0 ; SUBTRACT ASCII 0
900 23F5 3021 BMI GN4 ; BR IF NOT NUM
901 23F7 C90A CMP #10
902 23F9 B01D BCS GN4 ; BR IF NOT NUM
903 23FB 201A24 JSR GN5 ; OLD*2
904 23FE 6544 ADC CNUM ; PLUS NEW
905 2400 AA TAX
906 2401 A900 LDA #0
907 2403 6545 ADC CNUM+1
908 2405 A8 TAY
909 2406 201A24 JSR GN5 ; OLD*4
910 2409 201A24 JSR GN5 ; OLD*8
911 240C 8A TXA ; OLD*8 + OLD*2 + NEW
912 240D 6544 ADC CNUM
913 240F 8544 STA CNUM ; =OLD*10 + NEW
914 2411 98 TYA
915 2412 6545 ADC CNUM+1
916 2414 8545 STA CNUM+1
917 2416 90CF BCC GN2
918
919 GN4
920 2418 38 SEC
921 2419 60 RTS ; DONE
922 GN5
923 241A 0644 ASL CNUM ; CNUM * 2
924 241C 2645 ROL CNUM+1
925 241E B0FB BCS GN4
926 2420 60 RTS
927

```


PAGE

```

928
929
930
931      EPR
932 2421 A544      LDA      CNUM      ; GET PORT
933 2423 2095FE    JSR      OUTPRT    ; GO DO IT
934 2426 38        SEC
935 2427 200422    JSR      LDREGS
936 242A 203124    JSR      EPRX
937 242D 207A29    JSR      MVDSW
938 2430 60        RTS
939 2431 6C3600    EPRX      JMP      (OUTSW)
940
941
942
943      EIN
944 2434 A544      LDA      CNUM      ; GET PORT
945 2436 208BFE    JSR      INPRT     ; GO DO IT
946 2439 38        SEC
947 243A 200422    JSR      LDREGS
948 243D 204424    JSR      EINX
949 2440 206529    JSR      MVISW
950 2443 60        RTS
951 2444 6C3800    EINX      JMP      (INSW)
952
953
954
955      EMON
956 2447 AD272B    LDA      MONMOD    ; GET CURRENT BITS
957 244A 0D3A2B    ORA      IMBITS    ; OR IN NEW BITS
958 244D 8D272B    STA      MONMOD    ; SET NEW MODE
959 2450 60        RTS
960
961
962
963      ENOMON
964 2451 2C3A2B    BIT      IMBITS
965 2454 5003      BVC      ENM1
966 2456 201322    JSR      PRCRIF
967
968 2459 A970      LDA      #$70
969 245B 4D3A2B    EOR      IMBITS    ; INVERT INPUT BITS
970 245E 2D272B    AND      MONMOD    ; AND WITH CURRENT
971 2461 8D272B    STA      MONMOD    ; SET NEW MODE
972 2464 60        RTS
973

```


974			PAGE	
975			EMAXF - EXECUTE MAX FILES	
976				
977		EMAXF		
978	2465	A900	LDA	#0 ; RESET EXECUTE
979	2467	8D7D2B	STA	ESTATE
980	246A	A544	LDA	CNUM ; SAVE NEW NO FILES
981	246C	4B	PHA	
982	246D	202625	JSR	CLALL ; GO CLOSE ALL FILES
983	2470	6B	PLA	
984	2471	8D202B	STA	CNFTBS ; SET NEW NO FILE TBLS
985	2474	20E72B	JSR	BLDFTB ; GO BUILD NEW ONES
986	2477	60	RTS	
987				
988			EDEL - DELETE A FILE	
989				
990		EDEL		
991	247B	A905	LDA	#CRQDEL ; DELETE REQUEST
992	247A	20B824	JSR	OPEN ; GO OPEN
993	247D	207A2B	JSR	FILSRC ; FIND FILE
994	2480	A000	LDY	#0
995	2482	9B	TYA	
996	2483	9140	STA	(ZPGWRK),Y ; RESET FN
997	2485	60	RTS	
998				
999			ELOCK - LOCK A FILE	
1000				
1001		ELOCK		
1002	2486	A907	LDA	#CRQLCK ; SET LOCK
1003	2488	D002	BNE	ELGO
1004				
1005			EUNLK - UNLOCK A FILE	
1006				
1007		EUNLK		
1008	248A	A90B	LDA	#CRQUNL ; SET UNLOCK
1009		ELGO		
1010	248C	20B824	JSR	OPEN ; OPEN FILE & UNLOCK
1011	248F	20F924	JSR	ECLOSE ; CLOSE IT
1012	2492	60	RTS	; DONE
1013				

1014				PAGE	
1015				EREN - RENAME A FILE	
1016					
1017		EREN			
1018	2493	AD0820	LDA	FN2ADR	; MOVE FILE NAME2
1019	2496	BD8C35	STA	CCBFN2	
1020	2499	AD0920	LDA	FN2ADR+1	
1021	249C	BD8D35	STA	CCBFN2+1	
1022	249F	A909	LDA	#CRQRNM	
1023	24A1	BD292B	STA	TEMP1A	; SET RENAME
1024	24A4	20D624	JSR	EO3	; GO OPEN AND RENAME
1025	24A7	20F924	JSR	ECLOSE	; GO CLOSE
1026	24AA	60	RTS		; DONE
1027					
1028				EAPND - OPEN FILE FOR APPEND	
1029					
1030		EAPND			
1031	24AB	20B624	JSR	EOPEN	; GO OPEN
1032		AP1			
1033	24AE	20BB27	JSR	RBYTE	; READ A BYTE
1034	24B1	D0FB	BNE	AP1	; BR IF NOT ZERO
1035					
1036	24B3	4CE926	JMP	RWP3	; GO RE-POSITION
1037					

		PAGE		
1038				
1039			EOPEN - OPEN A FILE	
1040				
1041		EOPEN		
1042	24B6 A901	LDA	#CRGQPN	
1043		OPEN		
1044	24B8 8D292B	STA	TEMP1A	
1045	24BB AD322B	LDA	CL	; IF NO LENGTH ENTERED
1046	24BE D00A	BNE	EO1	; THEN SET DEFAULT OF 1
1047	24C0 AD332B	LDA	CL+1	
1048	24C3 D005	BNE	EO1	
1049	24C5 A901	LDA	#1	
1050	24C7 8D322B	STA	CL	
1051		EO1		
1052	24CA AD322B	LDA	CL	; MOVE REC LENGTH
1053	24CD 8DB835	STA	CCBRLN	
1054	24D0 AD332B	LDA	CL+1	
1055	24D3 8DB835	STA	CCBRLN+1	
1056		EO3		
1057	24D6 20F924	JSR	ECLOSE	; GO CLOSE IF OPEN
1058		EO4		
1059	24D9 A545	LDA	CNUM+1	; GET AVAIL ENTRY
1060	24DB D003	BNE	EO5	; BR IF ONE AVAIL
1061	24DD 4CE627	JMP	ENFA	; DONE - NO FILES AVAIL
1062		EO5		
1063	24E0 8541	STA	ZPGWRK+1	; MOVE AVAIL SLOT TO ZPG
1064	24E2 A544	LDA	CNUM	
1065	24E4 8540	STA	ZPGWRK	
1066		EO6		
1067	24E6 205928	JSR	MVFN1	; GO MOVE FILE NAME
1068	24E9 206428	JSR	MVBUPP	; GO MOVE BUF PTRS
1069	24EC 202628	JSR	OPNSUP	; GO SET UP OPEN
1070	24EF AD292B	LDA	TEMP1A	; SET OPEN REQ
1071	24F2 8DBA35	STA	CCBREQ	
1072	24F5 20CC27	JSR	DOSGO	; GO OPEN
1073	24F8 60	RTS		; DONE
1074				

		PAGE		
1075				
1076				ECLOSE - EXECUTE CLOSE FILE COMMAND
1077				
1078			ECLOSE	
1079	24F9 AD3B2B	LDA	FNAME1	
1080	24FC C9A0	CMP	##A0	
1081	24FE F026	BEQ	CLALL	
1082	2500 207A2B	JSR	FILSRC	; GO FIND FILE
1083	2503 B006	BCS	ECL1	; BR IF NOT FOUND
1084	2505 200C25	JSR	CLOSE	; GO CLOSE
1085	2508 4CF924	JMP	ECLOSE	; GO SEE IF ANY MORE OPEN
1086	250B 60	ECL1	RTS	
1087				
1088				CLOSE - CLOSE A FILE
1089				
1090			CLOSE	
1091	250C 20C52B	JSR	TSTEXC	
1092	250F D005	BNE	CLX	
1093	2511 A900	LDA	#0	
1094	2513 8D7D2B	STA	ESTATE	
1095		CLX		
1096	2516 A000	LDY	#0	; CLEAR 1ST FN
1097	2518 9B	TYA		; CHAR TO ZERO
1098	2519 9140	STA	(ZPGWRK), Y	
1099	251B 20642B	JSR	MVBUFF	; MOVE BUFFER PTRS
1100	251E A902	LDA	#CRQCLS	; SET CLOSE
1101	2520 8DBA35	STA	CCBREQ	
1102	2523 4CCC27	JMP	DOSGD	; GO CLOSE
1103				
1104				CLALL - CLOSE ALL FILES
1105				
1106			CLALL	
1107	2526 20A82B	JSR	TSINIT	; GO INIT FILE SEARCH
1108	2529 D005	BNE	CL1	
1109			CLO	
1110	252B 20B02B	JSR	TSNXT	; NEXT ENTRY
1111	252E F010	BEQ	CL2	; BR IF NO MORE
1112			CL1	
1113	2530 20C52B	JSR	TSTEXC	
1114	2533 F0F6	BEQ	CLO	
1115	2535 20C02B	JSR	TSTOPN	; GO TEST OPEN
1116	2538 F0F1	BEQ	CLO	; BR NOT OPEN
1117	253A 200C25	JSR	CLOSE	; GO CLOSE
1118	253D 4C2625	JMP	CLALL	; START OVER
1119	2540 60	CL2	RTS	; DONE
1120				

1121				PAGE	
1122				EBSV - EXECUTE BINARY SAVE	
1123					
1124			EBSV		
1125	2541	A909	LDA	#A+L	; IF A&L
1126	2543	2D2B2B	AND	INOPTS	; NOT GIVEN
1127	2546	C909	CMP	#A+L	
1128	2548	F003	BEG	EBSV1	
1129	254A	4CE227	JMP	ESYNTAX	; THEN ERROR
1130			EBSV1		
1131	254D	A904	LDA	#4	; SET BINARY FILE
1132	254F	20C625	JSR	SV1	; GO OPEN & TEST
1133	2552	AD392B	LDA	CA+1	; OUTPUT ADR OF BLOCK
1134	2555	AC382B	LDY	CA	
1135	2558	20D125	JSR	SV2	
1136	255B	AD332B	LDA	CL+1	; GO OPEN AND TEST
1137	255E	AC322B	LDY	CL	
1138	2561	20D125	JSR	SV2	; OUTPUT LENGTH
1139	2564	AD392B	LDA	CA+1	; GET ADR GIVEN
1140	2567	AC382B	LDY	CA	
1141	256A	4CF025	JMP	SV3	; OUTPUT BLOCK
1142					
1143				EBLD - EXECUTE BINARY LOAD	
1144					
1145			EBLD		
1146	256D	A904	LDA	#4	; SET BINARY FILE
1147	256F	20C625	JSR	SV1	; GO OPEN & TEST
1148	2572	204F26	JSR	LD2	; GO GET ADR
1149	2575	AA	TAX		
1150	2576	AD2B2B	LDA	INOPTS	
1151	2579	2901	AND	#A	; IF ADR NOT GIVEN
1152	257B	D006	BNE	EBLD1	
1153	257D	8E382B	STX	CA	; THEN USE ADR FROM FILE
1154	2580	8C392B	STY	CA+1	
1155			EBLD1		
1156	2583	204F26	JSR	LD2	; GET LENGTH
1157	2586	AE382B	LDX	CA	; GET GIVEN ADR
1158	2589	AC392B	LDY	CA+1	
1159	258C	4C8026	JMP	LD3	; GO GET BLOCK
1160					

PAGE

```

1161      ;
1162      ;      ESAVE - EXECUTE SAVE REQUEST
1163      ;
1164      ESAVE
1165 258F AD802B      LDA      ASIBSW      ; IF IB THEN
1166 2592 F019        BEQ       EIBSV      ; GO TO IB SAVE
1167 2594 A902        LDA       #2         ; GET APPLESOFT PGM
1168 2596 20C625      JSR       SV1        ; GO OPEN AND TEST
1169      ;
1170 2599 38          SEC              ; BLOCK LENGTH
1171 259A A5AF        LDA       ASEOP      ; =EOP-SOP
1172 259C E567        SBC       ASSOP
1173 259E AB          TAY
1174 259F A5B0        LDA       ASEOP+1
1175 25A1 E568        SBC       ASSOP+1
1176 25A3 20D125      JSR       SV2        ; GO OUTPUT LENGTH
1177      ;
1178 25A6 A568        LDA       ASSOP+1    ; BLOCK ADR
1179 25A8 A467        LDY       ASSOP      ; =SOP
1180 25AA 4CF025      JMP       SV3        ; GO OUTPUT BLOCK
1181      ;
1182      EIBSV
1183 25AD A901        LDA       #1         ; SET IB PGM
1184 25AF 20C625      JSR       SV1        ; GO OPEN AND TEST
1185      ;
1186 25B2 38          SEC              ; BLOCK LENGTH
1187 25B3 A54C        LDA       IBHMEM     ; =HIMEM-SOP
1188 25B5 E5CA        SBC       IBSOP
1189 25B7 AB          TAY
1190 25B8 A54D        LDA       IBHMEM+1
1191 25BA E5CB        SBC       IBSOP+1
1192 25BC 20D125      JSR       SV2        ; GO OUTPUT LENGTH
1193      ;
1194 25BF A5CB        LDA       IBSOP+1    ; BLOCK ADR
1195 25C1 A4CA        LDY       IBSOP      ; =SOP
1196 25C3 4CF025      JMP       SV3        ; GO OUTPUT BLOCK
1197      ;
1198      SV1
1199 25C6 8D9135      STA       CCBFUC     ; SET PGM TYPE
1200      SV1A
1201 25C9 48          PHA              ; SAVE PGM TYPE
1202 25CA 20B624      JSR       EOPEN     ; GO OPEN FILE
1203 25CD 68          PLA              ; GET SAVE TYPE
1204 25CE 4CDA28      JMP       TSTFUC    ; GO CHECK
1205      ;
1206      SV2
1207 25D1 8C9035      STY       CCBBLN    ; SET BLOCK LENGTH
1208 25D4 8C9235      STY       CCBDAT    ; AND DATA BYTE
1209 25D7 8D9135      STA       CCBBLN+1
1210 25DA A904        LDA       #CRQWR    ; INDICATE WRITE
1211 25DC 8D8A35      STA       CCBREQ
1212 25DF A901        LDA       #CRMNBT   ; NEXT BYTE
1213 25E1 8D8B35      STA       CCBRQM
1214 25E4 20CC27      JSR       DOSGO     ; GO WRITE

```


1215	25E7	AD9135		LDA	CCBBLN+1	; OTHER BYTE TOO
1216	25EA	8D9235		STA	CCBDAT	
1217	25ED	4CCC27		JMP	DOSGO	
1218						
1219	25F0	8C9235	SV3	STY	CCBBBA	; SET BLOCK ADR
1220	25F3	8D9335		STA	CCBBBA+1	
1221	25F6	A902		LDA	#CRMNBL	; INDICATE BLOCK I/O
1222	25F8	8D8B35		STA	CCBRQM	
1223	25FB	20CC27		JSR	DOSGO	; GO DO IT
1224	25FE	4CF924		JMP	ECLOSE	; CLOSE FILE
1225						

		PAGE			
1226					
1227					
1228					
1229					
1230	2601 202625	JSR	CLALL		; GO CLOSE ALL
1231	2604 AD802B	LDA	ASIBSW		; IF IB THEN
1232	2607 F022	BEG	EIBL		; GO TO IB LOAD
1233	2609 A902	LDA	#2		
1234	260B 20C925	JSR	SV1A		; GO OPEN AND TEST
1235					
1236	260E 204F26	JSR	LD2		; GO GET BLOCK LENGTH
1237					
1238	2611 18	CLC			
1239	2612 6567	ADC	ASSOP		; ADD BLOCK LENGTH TO SOP
1240	2614 AA	TAX			
1241	2615 98	TYA			
1242	2616 6568	ADC	ASSOP+1		
1243					
1244	2618 C574	CMP	ASHM1+1		; IF BL+SOP >= HMEM
1245	261A B070	BCS	MFULL		; THEN WON'T FIT
1246					
1247					
1248	261C 85B0	STA	ASEOP+1		; SET NEW EOP ADR
1249	261E 856A	STA	ASEOP2+1		
1250	2620 86AF	STX	ASEOP		
1251	2622 8669	STX	ASEOP2		
1252	2624 A667	LDX	ASSOP		; GET ADR WHERE TO LOAD
1253	2626 A468	LDY	ASSOP+1		
1254	2628 4C8026	JMP	LD3		; GO LOAD
1255					
1256					
1257	262B A901	LDA	#1		; SET IB PGM
1258	262D 20C925	JSR	SV1A		; GO OPEN AND TEST
1259					
1260	2630 204F26	JSR	LD2		; GO GET BLOCK LENGTH
1261					
1262	2633 38	SEC			; HMEM - BLOCK LENGTH
1263	2634 A54C	LDA	IBHMEM		; IS NEW SOP
1264	2636 ED5B2B	SBC	FNAME2		
1265	2639 AA	TAX			
1266	263A A54D	LDA	IBHMEM+1		
1267	263C ED5C2B	SBC	FNAME2+1		
1268	263F 904B	BCC	MFULL		
1269	2641 AB	TAY			
1270					
1271	2642 C44B	CPY	IBLMEM+1		; IF NEW SOP <= LMEM
1272	2644 9046	BCC	MFULL		
1273	2646 F044	BEG	MFULL		
1274	2648 84CB	STY	IBSOP+1		; SET NEW SOP
1275	264A 86CA	STX	IBSOP		
1276	264C 4C8026	JMP	LD3		
1277					
1278					
1279	264F AD0820	LDA	FN2ADR		; MOVE ADR OF WHERE

1280	2652	8D9235	STA	CCBBBA	; TO PUT DATA TO
1281	2655	AD0920	LDA	FN2ADR+1	; CCB
1282	2658	8D9335	STA	CCBBBA+1	
1283	265B	A900	LDA	#0	
1284	265D	8D9135	STA	CCBBLN+1	; READ INTO
1285	2660	A902	LDA	#2	
1286	2662	8D9035	STA	CCBBLN	
1287	2665	A903	LDA	#CRQRD	; READ
1288	2667	8D8A35	STA	CCBREQ	
1289	266A	A902	LDA	#CRMNBL	; BLOCK
1290	266C	8D8B35	STA	CCBRQM	
1291	266F	20CC27	JSR	DOSGO	
1292	2672	AD5C2B	LDA	FNAME2+1	
1293	2675	8D9135	STA	CCBBLN+1	
1294	2678	A8	TAY		
1295	2679	AD5B2B	LDA	FNAME2	
1296	267C	8D9035	STA	CCBBLN	
1297	267F	60	RTS		
1298					
1299			LD3		
1300	2680	8E9235	STX	CCBBBA	; SET BLOCK ADR
1301	2683	8C9335	STY	CCBBBA+1	
1302	2686	20CC27	JSR	DOSGO	; GET BLOCK
1303	2689	4CF924	JMP	ECLOSE	; GO CLOSE FILE
1304					
1305			MFULL		
1306	268C	20F924	JSR	ECLOSE	; GO CLOSE FILE
1307	268F	4CEA27	JMP	MFERR	; AND GIVE ERR MSG
1308					

			PAGE	
1309				
1310				
1311				
1312				
1313	2692	200126	JSR	ELOAD ; LOAD PGM
1314	2695	201322	JSR	PRCRIF
1315	2698	A9AC	LDA	##AC
1316	269A	6C5020	JMP	(RUN)
1317				
1318				
1319				
1320				
1321	269D	A54A	LDA	IBLMEM ; RESET START OF VARS
1322	269F	85CC	STA	IBSOV
1323	26A1	A54B	LDA	IBLMEM+1
1324	26A3	85CD	STA	IBSOV+1
1325	26A5	6C4E20	JMP	(CHAIN)
1326				
1327				
1328				
1329				
1330	26AB	200126	JSR	ELOAD ; LOAD PGM
1331	26AB	201322	JSR	PRCRIF
1332	26AE	A9AC	LDA	##AC ; FOR APPLE SOFT
1333	26B0	6C4E20	JMP	(CHAIN)
1334				

		PAGE		
1335				
1336				
1337				
1338				
1339	26B3 20C926	JSR	RWPSN	; GO POSITION FILE IF REQD
1340	26B6 A905	LDA	#5	
1341	26B8 8D1B2B	STA	OSTATE	; SET OSTATE=5
1342	26BB 4CE221	JMP	CERTN	; DONE
1343				
1344				
1345				
1346				
1347	26BE 20C926	JSR	RWPSN	; GO POSITION FILE IF REQD
1348	26C1 A901	LDA	#1	
1349	26C3 8D1A2B	STA	ISTATE	; SET I STATE = DISK INPUT
1350	26C6 4CE221	JMP	CERTN	; DONE
1351				
1352				
1353				
1354				
1355	26C9 207A2B	JSR	FILSRC	; FIND THE FILE
1356	26CC 9006	BCC	RWP1	; BR IF FILE FOUND
1357	26CE 20B624	JSR	EOPEN	; GO OPEN FOR KLUTZ
1358	26D1 4CD726	JMP	RWP2	; THEN SKIP NEXT LINE
1359				
1360	26D4 20642B	JSR	MVBUFP	; MOVE BUFF POINTERS
1361				
1362	26D7 AD2B2B	LDA	INOPTS	; GET IN OPTIONS
1363	26DA 2906	AND	#R+B	; WAS IT B OR R
1364	26DC F013	BEG	RWPR	; BR IF NOT
1365	26DE A203	LDX	#3	
1366	26E0 BD342B	LDA	CR, X	; MOVE REL REC
1367	26E3 9D8C35	STA	CCBRN, X	; AND REL BYTE
1368	26E6 CA	DEX		
1369	26E7 10F7	BPL	RWP2A	
1370				
1371	26E9 A90A	LDA	#CRQPOS	; INDICATE POISTION REQUEST
1372	26EB 8D8A35	STA	CCBREQ	
1373	26EE 20CC27	JSR	DOSGO	
1374	26F1 60	RWPR	RTS	; DONE
1375				

		PAGE		
1376				
1377			EEXEC - EXECUTE EXEC CMD	
1378				
1379		EEXEC		
1380	26F2 20B624	JSR	EOPEN	; OPEN FILE
1381	26F5 AD182B	LDA	CFTABA	; MOVE TABLE POINTERS
1382	26F8 8D7E2B	STA	EFTABA	
1383	26FB AD192B	LDA	CFTABA+1	
1384	26FE 8D7F2B	STA	EFTABA+1	
1385	2701 8D7D2B	STA	ESTATE	; SET EX STATE NON ZERO
1386	2704 D00E	BNE	EXP2	
1387				
1388				
1389			EPOS - EXECUTE POSITION	
1390				
1391		EPOS		
1392	2706 207A2B	JSR	FILSRC	
1393	2709 9006	BCC	EXP1	
1394	270B 20B624	JSR	EOPEN	
1395	270E 4C1427	JMP	EXP2	
1396	2711 20642B	JSR	MVBUFP	
1397				
1398	2714 AD2B2B	LDA	INOPTS	; GET OPTIONS
1399	2717 2904	AND	#R	; TEST R
1400	2719 F01B	BEQ	EX2	; BR NOT R
1401				
1402	271B AD342B	LDA	CR	; IF CR NOT ZERO
1403	271E D00B	BNE	EX1A	; THEN DECREMENT
1404	2720 AE352B	LDX	CR+1	
1405	2723 F011	BEQ	EX2	
1406	2725 CE352B	DEC	CR+1	
1407	2728 CE342B	DEC	CR	
1408	272B 20BB27	JSR	RBYTE	; AND READ A RCORD
1409	272E F06E	BEQ	ICFD4	
1410	2730 C98D	CMP	#8D	; UNTIL CR
1411	2732 D0F7	BNE	EX1	
1412	2734 F0E5	BEQ	EX0	; THEN TEST CR AGAIN
1413				
1414	2736 60	RTS		; DONE
1415				
1416			ECAT - PRINT CATALOG	
1417				
1418		ECAT		
1419	2737 AD222B	LDA	SVSTK	; SAVE SAVED STACK PTR FOR RESTORE
1420	273A 4B	PHA		
1421	273B A906	LDA	#CRQDIR	
1422	273D 20B824	JSR	OPEN	; GO PRETEND OPEN
1423	2740 6B	PLA		; GET SAVED STACK PTR
1424	2741 8D222B	STA	SVSTK	; RESTORE IT
1425	2744 60	RTS		
1426				

1427			PAGE	
1428			EAS - EXECUTE APPLESOFT REQUEST	
1429				
1430		EAS		
1431	2745	A900	LDA	#0
1432	2747	AE802B	LDX	ASIBSW ; IF NOT ALREADY IN AS
1433	274A	F006	BEQ	EAS0 ; THEN LOAD IT
1434	274C	8D802B	STA	ASIBSW ; ELSE CLEAR STATE
1435	274F	4C7620	JMP	DBINIT ; AND GO INIT IB
1436				
1437			EAS0	
1438	2752	20C522	JSR	CLRFNS
1439	2755	A206	LDX	#FASBL
1440	2757	BD802B	LDA	FASB-1, X ; MOVE SYSTEM FILE NAME
1441	275A	9D3A2B	STA	FNAME1-1, X
1442	275D	CA	DEX	
1443	275E	D0F7	BNE	EAS1
1444				
1445			EAS2	
1446	2760	A9C0	LDA	##C0
1447	2762	8D1A2B	STA	ISTATE ; FOR RAM APPLESOFT
1448	2765	4C9226	JMP	ERUN ; GO LOAD AND RUN
1449				
1450				
1451			EINIT - EXECUTE INIT COMMAND	
1452				
1453			EINIT	
1454	2768	AD0B20	LDA	ASTART+1
1455	276B	8D8B35	STA	CCBBSA
1456	276E	A90B	LDA	#CRQFMT
1457	2770	20B824	JSR	OPEN
1458	2773	4C8F25	JMP	ESAVE
1459				

		PAGE		
1460				
1461				
1462				
1463				
1464	2776 AD252B	LDA	SVA	; CHAR IN SAVED ACU
1465	2779 BD9235	STA	CCBDAT	; PUT INTO CCBDATA AREA
1466	277C A904	LDA	#CRQWR	; SET WRITE
1467	277E BD8A35	STA	CCBREQ	
1468	2781 A901	LDA	#CRMNBT	; SET NEXT BYTE
1469	2783 BD8B35	STA	CCBRQM	
1470	2786 20CC27	JSR	DOSGO	; GO WRITE BYTE
1471	2789 60	RTS		; RETURN TO CALLER
1472				
1473				
1474				
1475				
1476	278A A906	LDA	#6	; SET OUT STE = 6
1477				
1478	278C BD1B2B	STA	DSTATE	; TO CATCH ECHO
1479	278F 20BB27	JSR	RBYTE	
1480	2792 D00F	BNE	ICFD1	; BR IF NOT ZERO CHAR
1481				
1482	2794 A903	LDA	#3	
1483	2796 4D1B2B	EOR	DSTATE	
1484	2799 D003	BNE	ICFD4	
1485	279B 200C25	JSR	CLOSE	
1486				
1487	279E A905	LDA	#CREEOF	
1488	27A0 4CF027	JMP	ERROR	; GO TO ERROR
1489				
1490	27A3 BD252B	STA	SVA	; PUT INTO SAVED ACU
1491	27A6 3B	SEC		; SET FOR TS
1492	27A7 4C0022	JMP	ORTN	; GO RESTORE REGS AND RTS
1493				

		PAGE	
1494			
1495			
1496			
1497			
		NXTEXC	
1498	27AA AD7F2B	LDA	EFTABA+1 ; MOVE TABLE ADR
1499	27AD 8541	STA	ZPGWRK+1 ; NO ZPG
1500	27AF AD7E2B	LDA	EFTABA
1501	27B2 8540	STA	ZPGWRK
1502	27B4 206428	JSR	MVBUPF ; GO MOVE PTRS
1503	27B7 A903	LDA	#3
1504	27B9 D0D1	BNE	ICFD3
1505			
1506			
1507			
1508			
		RBYTE	
1509	27BB A903	LDA	#CRQRD ; SET READ
1510	27BD 8D8A35	STA	CCBREQ
1511	27C0 A901	LDA	#CRMNBT ; SET NEXT BYTE
1512	27C2 8D8B35	STA	CCBRQM
1513	27C5 20CC27	JSR	DOSGO ; GO TO DOS
1514	27C8 AD9235	LDA	CCBDAT ; GET THE DATA BYTE
1515	27CB 60	RTS	
1516			

		PAGE		
1517				
1518				
1519				
1520				
1521	27CC 20C92B	JSR	DOSENT	; GO TO DOS
1522	27CF B001	BCS	DG1	; BR IF ERROR
1523	27D1 60	RTS		; DONE
1524				
1525				
1526	27D2 AD9435	LDA	CCBSTA	; *** ERROR ***
1527	27D5 C905	CMP	#CREEOF	; GET STATUS OF I/O
1528	27D7 D006	BNE	DG3	; EOF ?
1529	27D9 A200	LDX	#0	; BR IF NOT
1530	27DB 8E9235	STX	CCBDAT	; SET OTHER EIF
1531	27DE 60	RTS		; DONE
1532				
1533	27DF 4CF027	JMP	ERROR	; GO DO ERROR
1534				
1535				

		PAGE		
1536				
1537			ERROR ROUTINE	
1538				
1539	27E2 A90B	ESYNTAX	LDA	#CREFLK+1
1540	27E4 D00A		BNE	ERROR
1541	27E6 A90C	ENFA	LDA	#CREFLK+2
1542	27E8 D006		BNE	ERROR
1543	27EA A90E	MFERR	LDA	#CREFLK+4
1544	27EC D002		BNE	ERROR
1545	27EE A90D	ERNU1	LDA	#CREFLK+3
1546				
1547		ERROR		
1548	27F0 8D252B		STA	SVA ; SAVE MSG NUMBER
1549	27F3 20712B		JSR	CLRSTS
1550	27F6 A200		LDX	#0
1551	27F8 200E2B		JSR	EMPR ; GO OUTPUT
1552	27FB AE252B		LDX	SVA ; GET SAVE MSG
1553	27FE 200E2B		JSR	EMPR ; GO OUTPUT MSG
1554	2801 A20F		LDX	#CREFLK+5
1555	2803 200E2B		JSR	EMPR
1556	2806 AE252B		LDX	SVA
1557	2809 A903		LDA	#03
1558	280B 6C5220		JMP	(BREAK)
1559				
1560		EMPR		
1561	280E BD082B		LDA	EMDTB, X ; GET ITS DISPL
1562	2811 AA		TAX	; INTO X
1563		EMPR1		
1564	2812 8E292B		STX	TEMP1A ; SAVE DISPL
1565	2815 BD702A		LDA	EMSG, X ; GET MSG CHAR
1566	2818 48		PHA	; SAVE CHAR
1567	2819 0980		ORA	#\$80 ; SET MSB ON
1568	281B 201022		JSR	ORTN1 ; OUTPUT CHAR
1569	281E AE292B		LDX	TEMP1A ; GET INDEX
1570	2821 E8		INX	; INCREMENT IT
1571	2822 68		PLA	; RE-LOAD CHAR
1572	2823 10ED		BPL	EMPR1 ; BR IF MORE CHARS
1573	2825 60		RTS	; DONE
1574				

PAGE

```

1575      ;
1576      ;      OPNSUP - OPEN SET UP
1577      ;
1578      OPNSUP
1579 2826 AD2C2B      LDA      CV          ; VOLUME
1580 2829 8D8E35      STA      CCBVOL
1581 282C AD2E2B      LDA      CD          ; DRIVE
1582 282F 8D8F35      STA      CCBDRV
1583 2832 AD302B      LDA      CS          ; SLOT
1584 2835 8D9035      STA      CCBSLT
1585 2838 AD0620      LDA      FN1ADR      ; FILENAME 1 ADR
1586 283B 8D9235      STA      CCBFN1
1587 283E AD0720      LDA      FN1ADR+1
1588 2841 8D9335      STA      CCBFN1+1
1589 2844 A540        LDA      ZPGWRK
1590 2846 8D182B      STA      CFTABA
1591 2849 A541        LDA      ZPGWRK+1
1592 284B 8D192B      STA      CFTABA+1
1593 284E AD9135      LDA      CCBFUC      ; IF NO CODE SET
1594 2851 D002        BNE      DS1
1595 2853 A940        LDA      #$40      ; THEN SET DATA
1596      DS1
1597 2855 8D9135      STA      CCBFUC
1598 2858 60          RTS
1599      ;
1600      ;      MVFN1 - MOVE FILE NAME 1 TO FILE PTR
1601      ;
1602      MVFN1
1603 2859 A01F        LDY      #31
1604 285B B93B2B      MVFN1A  LDA      FNAME1,Y
1605 285E 9140        STA      (ZPGWRK),Y
1606 2860 88          DEY
1607 2861 10F8        BPL      MVFN1A
1608 2863 60          RTS
1609      ;
1610      ;      MVBUFP - MOVE BUFFER PTRS TO CCB
1611      ;
1612      MVBUFP
1613 2864 A020        LDY      #32
1614 2866 B140        MVBP1  LDA      (ZPGWRK),Y
1615 2868 997635      STA      CCBFCB-32,Y
1616 286B CB          INY
1617 286C C02B        CPY      #40
1618 286E D0F6        BNE      MVBP1
1619 2870 60          RTS
1620      ;
1621      ;      CLRSTS - CLEAR STATES
1622      ;
1623      CLRSTS
1624 2871 A000        LDY      #0
1625 2873 8C1A2B      STY      ISTATE
1626 2876 8C1B2B      STY      OSTATE
1627 2879 60          RTS
1628

```


		PAGE		
1629				
1630			FILSRC - SEARCH FOR FILE NAME1	
1631				
1632		FILSRC		
1633	287A A900	LDA	#0	; CLEAR SV AVAIL
1634	287C 8545	STA	CNUM+1	
1635				
1636	287E 20A828	JSR	TSINIT	; GO INIT SEARCH
1637	2881 4C8928	JMP	FLS1A	
1638	2884 20B028	JSR	TSNXT	; LOOK AT NEXT
1639	2887 F01D	BEQ	FLS4	; BR IF NO NEXT
1640				
1641	2889 20C028	JSR	TSTOPN	; GO TEST OPEN
1642	288C D00A	BNE	FLS2	; BR IF OPEN
1643				
1644	288E A540	LDA	ZPGWRK	; SAVE AVAIL ENTRY ADR
1645	2890 8544	STA	CNUM	
1646	2892 A541	LDA	ZPGWRK+1	
1647	2894 8545	STA	CNUM+1	
1648	2896 D0EC	BNE	FLS1	; GO LOOK SOME MORE
1649				
1650	2898 A01F	LDY	#31	; FILE HAD 32 CHARTS
1651	289A B140	LDA	(ZPGWRK), Y	; GET CHAR
1652	289C D93B2B	CMP	FNAME1, Y	TEST CHAR
1653	289F D0E3	BNE	FLS1	; BR NOT
1654	28A1 88	DEY		
1655	28A2 10F6	BPL	FLS3	; LOOK AT 32 CHARS
1656	28A4 1B	CLC		; FOUND
1657	28A5 60	RTS		; DONE
1658				
1659	28A6 38	FLS4	SEC	; NOT FOUND
1660	28A7 60	RTS		; DONE
1661				

		PAGE		
1662				
1663				TSINIT - INITIALIZE FOR FTAB SEARCH
1664				TSNXT - GET NEXT FTAB ENTRY
1665				
1666		TSINIT		
1667	28A8 AD0020	LDA	FTAB	; GET 1ST PTR ADR
1668	28AB AE0120	LDX	FTAB+1	
1669	28AE D00A	BNE	TSST	
1670		TSNXT		
1671	28B0 A027	LDY	#39	; GET LINK
1672	28B2 B140	LDA	(ZPGWRK), Y	
1673	28B4 F009	BEG	TSR	; BR IF NO LINK
1674				
1675	28B6 AA	TAX		
1676	28B7 88	DEY		
1677	28B8 B140	LDA	(ZPGWRK), Y	
1678		TSST		
1679	28BA 8641	STX	ZPGWRK+1	
1680	28BC 8540	STA	ZPGWRK	
1681	28BE 8A	TXA		; SET NE CC
1682	28BF 60	TSR	RTS	; RTN
1683				
1684				TSTOPN - TST FOR OPEN FILE
1685				
1686		TSTOPN		
1687	28C0 A000	LDY	#0	; GET 1ST CHAR OF FN
1688	28C2 B140	LDA	(ZPGWRK), Y	
1689	28C4 60	RTS		
1690				
1691				TSTEXC - TEST CURRENT FILE FOR EXECUTE
1692				
1693		TSTEXC		
1694	28C5 AD7D2B	LDA	ESTATE	; IF ESTATE = 0
1695	28C8 F00E	BEG	TXC1	; THEN NO EXECUTE FILE
1696	28CA AD7E2B	LDA	EFTABA	; TEST CURRENT
1697	28CD C540	CMP	ZPGWRK	
1698	28CF D008	BNE	TXC2	; IS NOT
1699	28D1 AD7F2B	LDA	EFTABA+1	
1700	28D4 C541	CMP	ZPGWRK+1	
1701	28D6 F001	BEG	TXC2	; IS
1702	28D8 CA	TXC1	DEX	; IS NOT
1703	28D9 60	TXC2	RTS	; DONE
1704				

1705			PAGE
1706			TSTFUC - TEST FILE USE CODE FOR PGM
1707			
1708		TSTFUC	
1709	28DA 4D9135	EOR	CCBFUC
1710	28DD F007	BEQ	TFUCR
1711	28DF 297F	AND	##7F
1712	28E1 F003	BEQ	TFUCR
1713	28E3 4CEE27	JMP	ERNU1
1714	28E6 60	TFUCR	RTS
1715			

PAGE

```

1716      ;
1717      ; BLDFTB - BUILD FILE TABLES
1718      ; TABLE MAP:
1719      ; HIMEM, SOP
1720      ; SBUFF N (256)
1721      ; DBUFF N (256)
1722      ; FTB N (FCBLEN)
1723      ; HEADER N (40)
1724      ;
1725      ;
1726      ; SBUFF 1
1727      ; DBUFF 1
1728      ; FTB 1
1729      ; HEADER 1
1730      ; THIS PROGRAM
1731      ;
1732      ; HEADER MAP:
1733      ; FILENAME (32)
1734      ; FTB PTR (2)
1735      ; DBUF PTR (2)
1736      ; SBUF PTR (2)
1737      ; LINK (2)
1738      ;
1739      ; BLDFTB
1740 28E7 38 SEC
1741 28E8 AD0020 LDA FTAB ; START OF FTAB AREA
1742 28EB 8540 STA ZPGWRK ; IS 1ST FTB PTR
1743 28ED AD0120 LDA FTAB+1 ; HEADER
1744 28F0 8541 STA ZPGWRK+1
1745 28F2 AD202B LDA CNFTBS ; MOVE NO FTABS
1746 28F5 8D292B STA TEMP1A ; TO TEMP
1747      ;
1748 28F8 A000 BFT1 LDY #0
1749 28FA 98 TYA
1750 28FB 9140 STA (ZPGWRK),Y ; 1ST CHAR FN=0
1751 28FD A020 LDY #32 ; INC Y TO FCB PTR
1752 28FF 38 SEC
1753 2900 A540 LDA ZPGWRK ; END OF PTR HEADER
1754 2902 E92A SBC #FCBLEN ; MINUS FTAB LENGTH
1755 2904 9140 STA (ZPGWRK),Y ; IS START OF FTB
1756 2906 4B PHA ; SAVE LOW ADR BYTE
1757 2907 A541 LDA ZPGWRK+1
1758 2909 E900 SBC #0
1759 290B C8 INY
1760 290C 9140 STA (ZPGWRK),Y
1761 290E AA TAX
1762 290F CA DEX ; FTB ADR - 256
1763 2910 6B PLA ; IS ADR DIR BUFF
1764 2911 4B PHA
1765 2912 C8 INY
1766 2913 9140 STA (ZPGWRK),Y ; SET DIR BUF PTR
1767 2915 BA TXA
1768 2916 C8 INY
1769 2917 9140 STA (ZPGWRK),Y

```


1770	2919	AA	TAX		
1771	291A	CA	DEX		; DIR BUFF - 256
1772	291B	68	PLA		; IS SBUFF ADR
1773	291C	48	PHA		
1774	291D	CB	INY		
1775	291E	9140	STA	(ZPGWRK), Y	
1776	2920	CB	INY		
1777	2921	8A	TXA		
1778	2922	9140	STA	(ZPGWRK), Y	
1779					
1780	2924	CE292B	DEC	TEMP1A	; DECREMENT TABLE INDEX
1781	2927	F017	BEG	BFT2	; COUNT AND BR IF DONE
1782	2929	AA	TAX		
1783	292A	68	PLA		
1784	292B	38	SEC		
1785	292C	E928	SBC	#40	; SBUFF ADR - 40
1786	292E	CB	INY		
1787	292F	9140	STA	(ZPGWRK), Y	; IF ADR OF NEXT TAB
1788	2931	48	PHA		; WHICH GOES INTO
1789	2932	8A	TXA		; LINK
1790	2933	E900	SBC	#0	
1791	2935	CB	INY		
1792	2936	9140	STA	(ZPGWRK), Y	
1793	2938	B541	STA	ZPGWRK+1	; AND INTO ZPGWRK
1794	293A	68	PLA		; FOR NEXT ENTRY
1795	293B	B540	STA	ZPGWRK	; BUILD
1796	293D	4CF828	JMP	BFT1	; GO BUILD NEXT
1797					
1798					
1799	2940	48	PHA		
1800	2941	A900	LDA	#0	; SET LAST LINK
1801	2943	CB	INY		; TO ZERO
1802	2944	9140	STA	(ZPGWRK), Y	
1803	2946	CB	INY		
1804	2947	9140	STA	(ZPGWRK), Y	
1805					
1806	2949	AD802B	LDA	ASIBSW	; IF IB THEN GO
1807	294C	F00B	BEG	BFTIB	; DO IB STUFF
1808					
1809	294E	68	PLA		; SET APPLESOF
1810	294F	B574	STA	ASHM1+1	; UPPER MEM LIMITS
1811	2951	B570	STA	ASHM2+1	
1812	2953	68	PLA		
1813	2954	B573	STA	ASHM1	
1814	2956	B56F	STA	ASHM2	
1815	2958	60	RTS		
1816					
1817					
1818	2959	68	PLA		; SET IB
1819	295A	B54D	STA	IBHMEM+1	; UPPER MEM LIMITS
1820	295C	B5CB	STA	IBSOP+1	
1821	295E	68	PLA		
1822	295F	B54C	STA	IBHMEM	
1823	2961	B5CA	STA	IBSOP	
1824	2963	60	RTS		

1825 2964 60
1826

RTS

; DONE

1827 ; PAGE
1828 ; MVISW - MOVE INPUT SWITCH
1829 ;
1830 MVISW
1831 2965 A538 LDA INSW ; SAVE CHAR IN SWITCH
1832 2967 8D1E2B STA SVINS
1833 296A A539 LDA INSW+1
1834 296C 8D1F2B STA SVINS+1
1835 ;
1836 296F AD0220 LDA CINA ; SET DB CHAR IN ADR
1837 2972 8538 STA INSW
1838 2974 AD0320 LDA CINA+1
1839 2977 8539 STA INSW+1
1840 ;
1841 2979 60 RTS
1842 ;
1843 ; MVOSW - MOVE OUTPUT SWITCH
1844 ;
1845 MVOSW
1846 297A A536 LDA OUTSW ; SAVE CHAR OUT SWITCH
1847 297C 8D1C2B STA SVOUS
1848 297F A537 LDA OUTSW+1
1849 2981 8D1D2B STA SVOUS+1
1850 ;
1851 2984 AD0420 LDA COUTA ; SET DB CHAR OUT ADR
1852 2987 8536 STA OUTSW
1853 2989 AD0520 LDA COUTA+1
1854 298C 8537 STA OUTSW+1
1855 298E 60 RTS
1856

		PAGE	
1857			
1858			COMMAND NAME TABLE
1859			
1860		EC1	
1861		CMDNTB	
1862	298F 49	DB01	"INIT"
	2990 4E		
	2991 49		
	2992 D4		
1863	2993 4C	DB01	"LOAD"
	2994 4F		
	2995 41		
	2996 C4		
1864	2997 53	DB01	"SAVE"
	2998 41		
	2999 56		
	299A C5		
1865	299B 52	DB01	"RUN"
	299C 55		
	299D CE		
1866	299E 43	DB01	"CHAIN"
	299F 48		
	29A0 41		
	29A1 49		
	29A2 CE		
1867	29A3 44	DB01	"DELETE"
	29A4 45		
	29A5 4C		
	29A6 45		
	29A7 54		
	29A8 C5		
1868	29A9 4C	DB01	"LOCK"
	29AA 4F		
	29AB 43		
	29AC CB		
1869	29AD 55	DB01	"UNLOCK"
	29AE 4E		
	29AF 4C		
	29B0 4F		
	29B1 43		
	29B2 CB		
1870	29B3 43	DB01	"CLOSE"
	29B4 4C		
	29B5 4F		
	29B6 53		
	29B7 C5		
1871	29B8 52	DB01	"READ"
	29B9 45		
	29BA 41		
	29BB C4		
1872	29BC 45	DB01	"EXEC"
	29BD 58		
	29BE 45		
	29BF C3		

1873	29C0 57	DB01	"WRITE"
	29C1 52		
	29C2 49		
	29C3 54		
	29C4 C5		
1874	29C5 50	DB01	"POSITION"
	29C6 4F		
	29C7 53		
	29C8 49		
	29C9 54		
	29CA 49		
	29CB 4F		
	29CC CE		
1875	29CD 4F	DB01	"OPEN"
	29CE 50		
	29CF 45		
	29D0 CE		
1876	29D1 41	DB01	"APPEND"
	29D2 50		
	29D3 50		
	29D4 45		
	29D5 4E		
	29D6 C4		
1877	29D7 52	DB01	"RENAME"
	29D8 45		
	29D9 4E		
	29DA 41		
	29DB 4D		
	29DC C5		
1878	29DD 43	DB01	"CATALOG"
	29DE 41		
	29DF 54		
	29E0 41		
	29E1 4C		
	29E2 4F		
	29E3 C7		
1879	29E4 4D	DB01	"MON"
	29E5 4F		
	29E6 CE		
1880	29E7 4E	DB01	"NOMON"
	29E8 4F		
	29E9 4D		
	29EA 4F		
	29EB CE		
1881	29EC 50	DB01	"PR#"
	29ED 52		
	29EE A3		
1882	29EF 49	DB01	"IN#"
	29F0 4E		
	29F1 A3		
1883	29F2 4D	DB01	"MAXFILES"
	29F3 41		
	29F4 5B		
	29F5 46		
	29F6 49		

	29F7 4C		
	29F8 45		
	29F9 D3		
1884	29FA 41	DB01	"APPLESOFT"
	29FB 50		
	29FC 50		
	29FD 4C		
	29FE 45		
	29FF 53		
	2A00 4F		
	2A01 46		
	2A02 D4		
1885	2A03 42	DB01	"BSAVE"
	2A04 53		
	2A05 41		
	2A06 56		
	2A07 C5		
1886	2A08 42	DB01	"BLOAD"
	2A09 4C		
	2A0A 4F		
	2A0B 41		
	2A0C C4		
1887	2A0D 00	DB	O
1888			

		PAGE		
1889				
1890				COMMAND SYNTAX OP EQUATES FOR SYNTAX BYTE ONE
1891				
1892	0080	NPB	EQU	\$80 ; NO PARMS OK, COMMAND GOES TO BASIC
1893	0040	NPE	EQU	\$40 ; NO PARMS OK, COMMAND TO EXECUTION RTN
1894	0020	FN1	EQU	\$20 ; FILE NAME1 REGD
1895	0010	FN2	EQU	\$10 ; FILE NAME2 REGD
1896	0008	NUM1	EQU	\$08 ; NUMERIC 0-7 REGD
1897	0004	NUM2	EQU	\$04 ; NUMERIC 1-10 REGD
1898				
1899				COMMAND SYNTAX OP EQUATES FOR SYNTAX BYTE TWO
1900				
1901	0040	V	EQU	\$40 ; VOLUME ALLOWED
1902	0020	D	EQU	\$20 ; DRIVE ALLOWED
1903	0010	S	EQU	\$10 ; SLOT ALLOWED
1904	0008	L	EQU	\$08 ; LENGTH ALLOWED
1905	0004	R	EQU	\$04 ; RECORD NUMBER ALLOWED
1906	0002	B	EQU	\$02 ; BYTE NUMBER ALLOWED
1907	0001	A	EQU	\$01 ; ADDRESS
1908	0080	CID	EQU	\$80 ; C, I, OR O ALLOWED
1909				
1910				COMMAND SYNTAX TABLE
1911				EACH COMMAND HAS TWO BYTE ENTRY
1912				
1913		CMDSTB		
1914	2A0E 20		DB	FN1, V+D+S ; INIT
	2A0F 70			
1915	2A10 A0		DB	NPB+FN1, V+D+S ; LOAD
	2A11 70			
1916	2A12 A0		DB	NPB+FN1, V+D+S ; SAVE
	2A13 70			
1917	2A14 A0		DB	NPB+FN1, V+D+S ; RUN
	2A15 70			
1918	2A16 20		DB	FN1, V+D+S ; CHAIN
	2A17 70			
1919	2A18 20		DB	FN1, V+D+S ; DELETE
	2A19 70			
1920	2A1A 20		DB	FN1, V+D+S ; LOCK
	2A1B 70			
1921	2A1C 20		DB	FN1, V+D+S ; UNLOCK
	2A1D 70			
1922	2A1E 60		DB	NPE+FN1, 0 ; CLOSE
	2A1F 00			
1923	2A20 20		DB	FN1, B+R ; READ
	2A21 06			
1924	2A22 20		DB	FN1, R+V+D+S ; EXEC
	2A23 74			
1925	2A24 20		DB	FN1, B+R ; WRITE
	2A25 06			
1926	2A26 20		DB	FN1, R ; POSITION
	2A27 04			
1927	2A28 20		DB	FN1, L+V+D+S ; OPEN
	2A29 78			
1928	2A2A 20		DB	FN1, L+V+D+S ; APPEND

1929	2A2B 78			
	2A2C 30	DB	FN1+FN2, V+D+S	; RENAME
	2A2D 70			
1930	2A2E 40	DB	NPE, V+D+S	; CATALOG
	2A2F 70			
1931	2A30 40	DB	NPE, CIO	; MONITOR
	2A31 80			
1932	2A32 40	DB	NPE, CIO	; NO MONITOR
	2A33 80			
1933	2A34 08	DB	NUM1, 0	; PR#
	2A35 00			
1934	2A36 08	DB	NUM1, 0	; IN#
	2A37 00			
1935	2A38 04	DB	NUM2, 0	; MAXFILES
	2A39 00			
1936	2A3A 40	DB	NPE, V+D+S	; APPLESOFT
	2A3B 70			
1937	2A3C 20	DB	FN1, V+D+S+A+L	; BSAVE
	2A3D 79			
1938	2A3E 20	DB	FN1, V+D+S+A	; BLOAD
	2A3F 71			
1939				

PAGE

1940 ;
 1941 ; OPTAB - OPTIONAL PARMS SYNTAX TABLES
 1942 ;

1943 OPTAB1

1944 2A40 D6 DB11 "VDSL RBACIO"

2A41 C4

2A42 D3

2A43 CC

2A44 D2

2A45 C2

2A46 C1

2A47 C3

2A48 C9

2A49 CF

1945 000A OPT1L EQU *-OPTAB1

1946 OPTAB2

1947 2A4A 40 DB V, D, S, L, R, B, A, CIO+MC, CIO+MI, CIO+MO

2A4B 20

2A4C 10

2A4D 08

2A4E 04

2A4F 02

2A50 01

2A51 C0

2A52 A0

2A53 90

1948 OPTAB3

1949 2A54 0000 DB @@0, @@254 ; VOL RANGE

2A56 FE00

1950 2A58 0100 DB @@1, @@2 ; DRIVE RANGE

2A5A 0200

1951 2A5C 0100 DB @@1, @@7 ; SLOT RANGE

2A5E 0700

1952 2A60 0100 DB @@1, @@32767 ; LENGTH RANGE

2A62 FF7F

1953 2A64 0000 DB @@0, @@32767 ; REC NO RANGE

2A66 FF7F

1954 2A68 0000 DB @@0, @@32767 ; REC BYTE NO RANGE

2A6A FF7F

1955 2A6C 0000 DB @@0, @@\$C000 ; ADDRESS RANGE

2A6E 00C0

1956

PAGE

1957

1958

1959

1960

EMSG

ERROR MESSAGE TABLES

1961 2A70 0D

2A71 07

DB

\$0D, \$07

1962 2A72 2A

2A73 2A

2A74 2A

2A75 44

2A76 49

2A77 53

2A78 4B

2A79 3A

2A7A A0

DB01

"***DISK: "

1963 000B

EM1

EQU

*-EMSG

1964 000B

EM2

EQU

*-EMSG

1965 000B

EM3

EQU

*-EMSG

1966 000B

EM4

EQU

*-EMSG

1967 2A7B 53

2A7C 59

2A7D D3

DB01

"SYS"

1968 000E

EM5

EQU

*-EMSG

1969 2A7E 45

2A7F 4E

2A80 44

2A81 20

2A82 4F

2A83 46

2A84 20

2A85 44

2A86 41

2A87 54

2A88 C1

DB01

"END OF DATA"

1970 0019

EM6

EQU

*-EMSG

1971 2A89 46

2A8A 49

2A8B 4C

2A8C 45

2A8D 20

2A8E 4E

2A8F 4F

2A90 54

2A91 20

2A92 46

2A93 4F

2A94 55

2A95 4E

2A96 C4

DB01

"FILE NOT FOUND"

1972 0027

EM7

EQU

*-EMSG

1973 2A97 56

2A98 4F

2A99 4C

2A9A 55

DB01

"VOLUME MISMATCH"

2A9B 4D

2A9C 45

2A9D 20

2A9E 4D

2A9F 49

2AA0 53

2AA1 4D

2AA2 41

2AA3 54

2AA4 43

2AA5 C8

1974 0036

EM8

EQU

*-EMSG

1975 2AA6 44

DB01

"DISK I/O"

2AA7 49

2AA8 53

2AA9 4B

2AAA 20

2AAB 49

2AAC 2F

2AAD CF

1976 003E

EM9

EQU

*-EMSG

1977 2AAE 44

DB01

"DISK FULL"

2AAF 49

2AB0 53

2AB1 4B

2AB2 20

2AB3 46

2AB4 55

2AB5 4C

2AB6 CC

1978 0047

EM10

EQU

*-EMSG

1979 2AB7 46

DB01

"FILE LOCKED"

2AB8 49

2AB9 4C

2ABA 45

2ABB 20

2ABC 4C

2ABD 4F

2ABE 43

2ABF 4B

2AC0 45

2AC1 C4

1980 0052

EM11

EQU

*-EMSG

1981 2AC2 43

DB01

"CMD SYNTAX"

2AC3 4D

2AC4 44

2AC5 20

2AC6 53

2AC7 59

2AC8 4E

2AC9 54

2ACA 41

2ACB DB

1982 005C

EM12

EQU

*-EMSG

1983 2ACC 4E

DB01

"NO FILE BUFFS AVAIL"

2ACD 4F

2ACE 20

2ACF 46

2AD0 49

2AD1 4C

2AD2 45

2AD3 20

2AD4 42

2AD5 55

2AD6 46

2AD7 46

2AD8 53

2AD9 20

2ADA 41

2ADB 56

2ADC 41

2ADD 49

2ADE CC

1984 006F

EM13

EQU

*-EMSG

1985 2ADF 4E

DB01

"NOT BASIC PROGRAM"

2AE0 4F

2AE1 54

2AE2 20

2AE3 42

2AE4 41

2AE5 53

2AE6 49

2AE7 43

2AEB 20

2AE9 50

2AEA 52

2AEB 4F

2AEC 47

2AED 52

2AEE 41

2AEF CD

1986 0080

EM14

EQU

*-EMSG

1987 2AF0 50

DB01

"PROGRAM TOO LARGE"

2AF1 52

2AF2 4F

2AF3 47

2AF4 52

2AF5 41

2AF6 4D

2AF7 20

2AF8 54

2AF9 4F

2AFA 4F

2AFB 20

2AFC 4C

2AFD 41

2AFE 52

2AFF 47

2B00 C5

1988

1989		0091	EML	EQU	*-EMSG
1990	2B01	20		DB	" ERROR "
	2B02	45			
	2B03	52			
	2B04	52			
	2B05	4F			
	2B06	52			
1991	2B07	8D		DB	\$8D
1992			EMDTB		
1993	2B08	00		DB	0, EM1, EM2, EM3, EM4
	2B09	0B			
	2B0A	0B			
	2B0B	0B			
	2B0C	0B			
1994	2B0D	0E		DB	EM5, EM6, EM7, EM8, EM9
	2B0E	19			
	2B0F	27			
	2B10	36			
	2B11	3E			
1995	2B12	47		DB	EM10, EM11, EM12, EM13, EM14
	2B13	52			
	2B14	5C			
	2B15	6F			
	2B16	80			
1996	2B17	91		DB	EML
1997					

PAGE

```

1998 ;
1999 ; MISC BUT REQD CELLS
2000 ;
2001 2B1B 0000 CFTABA DB @0 ; CURRENT FILE TABLE POINTER
2002 2B1A 00 ISTATE DB 0 ; INPUT STATE
2003 2B1B 00 OSTATE DB 0 ; OUTPUT STATE
2004 2B1C 0000 SVDUTS DB @0 ; SAVED OUT SWITCH
2005 2B1E 0000 SVINS DB @0 ; SAVED IN SWITCH
2006 2B20 00 CNFTBS DB 0 ; CURRENT NO FILE TABLES
2007 2B21 03 DFNFTB DB 3 ; DEFAULT NO FILE TABLES
2008 2B22 00 SVSTK DB 0 ; SAVED STACK PTR
2009 2B23 00 SVX DB 0 ; DSAVED X REG
2010 2B24 00 SVY DB 0 ; SAVED Y REG
2011 2B25 00 SVA DB 0 ; SAVED ACU
2012 2B26 00 LBUFD DB 0 ; LINE BUFF DISPL
2013 2B27 00 MONMOD DB 0 ; MONITOR MODE BITS
2014 0040 MC EQU $40 ; MONITOR CMDS
2015 0020 MI EQU $20 ; MONITOR INPUT
2016 0010 MO EQU $10 ; MONITOR OUTPUT
2017 2B28 FF CMDNO DB $FF ; COMMAND NO
2018 2B29 00 TEMP1A DB 0
2019 2B2A 00 TEMP2A DB 0
2020 2B2B 00 INOPTS DB 0 ; INPUT OPTIONS
2021 CUROPT ; CURRENT OPTIONS
2022 2B2C 0000 CV DB @@0 ; VOLUME
2023 2B2E 0000 CD DB @@0 ; DRIVE
2024 2B30 0000 CS DB @@0 ; SLOT
2025 2B32 0100 CL DB @@1 ; RECORD LENGTH
2026 2B34 0000 CR DB @@0 ; RECORD NUMBER
2027 2B36 0000 CB DB @@0 ; RECORD BYTE
2028 2B38 0000 CA DB @@0 ; ADDRESS
2029 2B3A 00 IMBITS DB 0
2030 2B3B FNAME1 RMB 32 ; FILENAME 1
2031 2B5B FNAME2 RMB 32 ; FILENAME 2
2032 2B7B 03 DFNFTS DB 3 ; DEFAULT FILE TABLES = 3
2033 2B7C 84 CCHAR DB $84 ; CONTROL CHAR
2034 2B7D 00 ESTATE DB 0 ; EXECUTE STATE
2035 2B7E 00 EFTABA DB 0,0 ; EXECUTE FILE TABLE POINTER
2036 2B7F 00
2036 2B80 00 ASIBSW DB 0 ; APPLESOFT, IB SWITCH
2037 2B81 D3 FASB DB11 "SYSASB"
2037 2B82 D9
2037 2B83 D3
2037 2B84 C1
2037 2B85 D3
2037 2B86 C2
2038 0006 FASBL EQU *-FASB
2039

```


PAGE

2040

2041

2042

2043

SAT2

2044 2B87 E837

AIOB

DB

@@IOB

; 5-ADR IOB

2045 2B89 8A33

AVTOC

DB

@@VTOC

; 6-ADR VTOC

2046 2B8B 8A34

AVOLDR

DB

@@VOLDIR

; 7-ADR VOLDIR

2047 2B8D 0040

AEND

DB

@@EDOS

FEND OF DOS

2048

;

2049 2B8F 5C33

CMDVT

DB

@@GOODIO-1

; 0-NULL

2050 2B91 E42B

DB

@@FOPEN-1

; 1-OPEN FILE

2051 2B93 952C

DB

@@FCLOSE-1

; 2-CLOSE FILE

2052 2B95 BF2C

DB

@@FREAD-1

; 3-READ DATA

2053 2B97 D72C

DB

@@FWRITE-1

; 4-WRITE DATA

2054 2B99 802D

DB

@@FDEL-1

; 5-DELETE FILE

2055 2B9B ED2D

DB

@@RDIR-1

; 6-READ DIRECTORY

2056 2B9D 572D

DB

@@FLOCK-1

; 7-LOCK A FILE

2057 2B9F 5E2D

DB

@@FUNLCK-1

; 8-UNLOCK A FILE

2058 2BA1 A12C

DB

@@FRNME-1

; 9-RENAME

2059 2BA3 7A2D

DB

@@FPOSTN-1

; 10-POSITION A FILE

2060 2BA5 6A2E

DB

@@FFMT-1

; FORMAT

2061 2BA7 5C33

DB

@@GOODIO-1

; 11-SPARE

2062

;

2063

RVT

2064 2BA9 5C33

DB

@@GOODIO-1

2065 2BAB F12C

DB

@@RNXYT-1

; 1-RD NEXT BYTE

2066 2BAD FD2C

DB

@@RNXYBLK-1

; 1-RD NEXT BLOCK

2067 2BAF EE2C

DB

@@RSPBYT-1

; 2-RD SPECIFIC BYTE

2068 2BB1 FA2C

DB

@@RSPBLK-1

; 3 - RD SPECIFIC BLOCK

2069 2BB3 5C33

DB

@@GOODIO-1

; 4 - SPARE

2070 2BB5 5C33

DB

@@GOODIO-1

; 5- SPARE

2071 2BB7 5C33

DB

@@GOODIO-1

; 6 - SPARE

2072

;

2073

WVT

2074 2BB9 5C33

DB

@@GOODIO-1

2075 2BBB 252D

DB

@@WNXYT-1

; 1-WR NEXT BYTE

2076 2BBD 312D

DB

@@WNXYBLK-1

; WR NEXT BLOCK

2077 2BBF 222D

DB

@@WSPBYT-1

; 2-WR SPECIFIC BYTE

2078 2BC1 2E2D

DB

@@WSPBLK-1

; 3-WR SPECIFIC BLOCK

2079 2BC3 5C33

DB

@@GOODIO-1

; 4 - SPARE

2080 2BC5 5C33

DB

@@GOODIO-1

; 5- SPARE

2081 2BC7 5C33

DB

@@GOODIO-1

; 6- SPARE

2082

EAT2

2083

PAGE

2084					
2085					
2086					
2087					
2088					
2089					
2090					
2091					
2092		SC2			
2093		DOSENT			
2094	2BC9	BA	TSX		
2095	2BCA	BE7F33	STX	ENTSTK	
2096	2BCD	204A2E	JSR	CLCFB	; GO CALCULATE FCB
2097	2BD0	ADBA35	LDA	CCBREQ	; GET REQUEST
2098	2BD3	C90C	CMP	#CRGMAX	; TTEST REQ RANGE
2099	2BD5	B00B	BCS	ERR2	; BR OUT OF RANGE
2100	2BD7	0A	ASLA		; REQ CODE *2
2101	2BD8	AA	TAX		
2102	2BD9	BD902B	LDA	CMDVT+1, X	; PUSH ADR ONTO STACK
2103	2BDC	4B	PHA		
2104	2BDD	BD8F2B	LDA	CMDVT, X	
2105	2BE0	4B	PHA		
2106	2BE1	60	DENRTS	RTS	
2107	2BE2	4C4133	ERR2	JMP	ERROR2
2108					

4600

2887

1479

		PAGE		
2109				
2110			FOPEN - OPEN A FILE	
2111				
2112			FOPEN	
2113	2BE5 20EB2B		JSR	DOPEN
2114	2BE8 4C5D33		JMP	GOODIO
2115				
2116			DOPEN	
2117				
2118	2BEB 20662C		JSR	DCBSUP
2119				
2120				
2121	2BEE A901		LDA	#1
2122	2BF0 8DB135		STA	DCBSDL+1
2123	2BF3 AE8D35		LDX	CCBRLN+1 ; MOVE RECORD LENGTH
2124	2BF6 AD8C35		LDA	CCBRLN
2125	2BF9 D005		BNE	F02
2126	2BFB E000		CPX	#0
2127	2BFD D001		BNE	F02
2128	2BFF E8		INX	; SET RL=256
2129	2C00 8DB635	F02	STA	DCBRCL
2130	2C03 8EB735		STX	DCBRCL+1
2131				
2132	2C06 20B631		JSR	FNDFIL ; GO FIND FILE
2133	2C09 9033		BCC	F03 ; BR IF FOUND
2134				; CREATE FILE
2135	2C0B 8E8033		STX	TEMP1 ; SAVE VDIR INDEX
2136	2C0E 202932		JSR	GETSEC ; GO ALLOCATE SECTOR
2137	2C11 AE8033		LDX	TEMP1
2138	2C14 9D9634		STA	VDFILE+1, X ; PUT SECTOR INTO VDIR
2139	2C17 8DA235		STA	DCBFDS ; PUT SECTOR AS 1ST FILE DIR
2140	2C1A 8DA435		STA	DCBCDS ; PUT SECTOR AS CURRENT FILE DIR
2141				
2142	2C1D ADBD35		LDA	DCBATK ; GET ALLOCATED TRACK
2143	2C20 9D9534		STA	VDFILE, X ; PUT INTO VDIR
2144	2C23 8DA135		STA	DCBFDT ; AND AS 1ST FILE DIR
2145	2C26 8DA335		STA	DCBCDT ; AND AS CURRENT FILE DIR
2146				
2147	2C29 AD9135		LDA	CCBFUC ; SET USE CODE
2148	2C2C 9D9734		STA	VDFILE+2, X ; INTO DIRECTORY
2149				
2150	2C2F 203430		JSR	WRVDIR ; GO WRITE VOL DIRECTORY
2151				
2152	2C32 20002F		JSR	MVFCBD ; MOVE FILE DIR ADR TO ZP
2153	2C35 20112F		JSR	CLRSEC ; GO CLEAR IT
2154	2C38 20372F		JSR	WRFDGO ; GO WRITE FILE DIRECTORY
2155				DONE CREATION
2156	2C3B AE8033		LDX	TEMP1 ; RE-GET INDEX
2157				
2158		F03		
2159	2C3E BD9534		LDA	VDFILE, X ; MOVE FILE DIR TRACK
2160	2C41 8DA135		STA	DCBFDT
2161	2C44 BD9634		LDA	VDFILE+1, X ; MOVE FILE DIR SECTOR
2162	2C47 8DA235		STA	DCBFDS


```

2163 2C4A BD9734      LDA      VDFILE+2,X      ; MOVE FILE USE CODE
2164 2C4D BD9135      STA      CCBFUC
2165 2C50 BDC235      STA      DCBFUC
2166
2167 2C53 A9FF        LDA      #255            ; INDICATE NO SECTOR
2168 2C55 BDAE35      STA      DCBCMS          ; IN MEMORY
2169 2C58 BDAF35      STA      DCBCMS+1
2170 2C5B ADB133      LDA      VTDMS          ; MOVE MAX FD SECTS
2171 2C5E BDA835      STA      DCBDMS          ; TO DCB
2172 2C61 1B          CLC
2173 2C62 205B2F      JSR      RDFDIR          ; READ 1ST DIRECTORY RECORD
2174
2175
2176
2177 2C65 60          RTS
2178
2179                DCBSUP
2180 2C66 A900        LDA      #0
2181 2C68 AA          TAX
2182 2C69 9DA135      F01     STA      FCBDCB,X      ; CLEAR DCB
2183 2C6C EB          INX
2184 2C6D E029        CPX      #DCBLEN
2185 2C6F D0F8        BNE      F01
2186
2187 2C71 ADBE35      LDA      CCBVOL          ; MOVE VOL
2188 2C74 49FF        EOR      #$FF          ; INVERT VOL BITS
2189 2C76 BDC535      STA      DCBVOL
2190 2C79 ADBF35      LDA      CCBDRV          ; MOVE DRIVE
2191 2C7C BDC435      STA      DCBDRV
2192 2C7F AE8333      LDX      ENTSLT          ; MOVE SLOT IF GIVEN
2193 2C82 AD9035      LDA      CCBSLT          ; GET USER SPEC SLOT
2194 2C85 F005        BEQ      F01A          ; BR IF NOT US SLOT
2195 2C87 0A          ASLA                     ; SLOT*16
2196 2C88 0A          ASLA
2197 2C89 0A          ASLA
2198 2C8A 0A          ASLA
2199 2C8B AA          TAX
2200                F01A
2201 2C8C BEC335      STX      DCBSLT
2202 2C8F AD7A33      LDA      GENVTN          ; MOVE VTOC TRACK NO
2203 2C92 BDC635      STA      DCBVTN
2204 2C95 60          RTS
2205

```


2206				PAGE	
2207				FCLOSE - CLOSE A FILE	
2208					
2209			FCLOSE		
2210	2C96	20A132	JSR	FRETRK	; FREE UNUSED SECTORS
2211	2C99	201A2F	JSR	WRSECT	; WRITE OPEN SECTOR
2212	2C9C	20312F	JSR	WRFDIR	; GO WRITE FILE DIRECTORY
2213					
2214					
2215					
2216	2C9F	4C5D33	JMP	GOODIO	; DONE
2217					

		PAGE			
2218					
2219					
2220			FRNME - RENAME A FILE		
2221			FRNME		
2222	2CA2	20EB2B	JSR	DOPEN	; GO OPEN FILE
2223	2CA5	ADC235	LDA	DCBFUC	; GET USE CODE
2224	2CA8	302B	BMI	ER10	; BR IF LOCKED
2225	2CAA	AD8C35	LDA	CCBFN2	; MOVE NEW FN
2226	2CAD	8542	STA	ZPGFCB	; PTR TO ZPG
2227	2CAF	AD8D35	LDA	CCBFN2+1	
2228	2CB2	8543	STA	ZPGFCB+1	
2229	2CB4	AE8033	LDX	TEMP1	; GET VDIR INDEX
2230	2CB7	200132	JSR	MVFN	; GO MOVE FILE NAME
2231	2CBA	203430	JSR	WRVDIR	; GO WRITE VDIR
2232	2CBD	4C5D33	JMP	GOODIO	; DONE RENAME
2233					


```

                PAGE
2234            ;
2235            ; FREAD - READ A FILE
2236            ;
2237            FREAD
2238            ;
2239 2CC0 AD8B35      LDA      CCBQRM      ; GET REQ MOD
2240 2CC3 C905        CMP      #CRMMAX    ; TEST LIMIT
2241 2CC5 B00B        BCS      ERR3A      ; BR BAD
2242            ;
2243 2CC7 0A          ASLA                     ; CODE*2
2244 2CC8 AA          TAX
2245 2CC9 BDAA2B      LDA      RVT+1,X    ; GET READ ROUTINE
2246 2CCC 4B          PHA                     ; VECTOR ADR
2247 2CCD BDA92B      LDA      RVT,X
2248 2CD0 4B          PHA                     ; AND
2249 2CD1 60          RTS                     ; GO TO IT
2250            ;
2251 2CD2 4C4533      ERR3A  JMP      ERROR3
2252 2CD5 4C5933      ER10  JMP      ERRR10
2253            ;
2254            ; FWRITE - WRITE A FILE
2255            ;
2256            FWRITE
2257 2CDB ADC235      LDA      DCBFUC      ; IS FILE LOCKED
2258 2CDB 30F8        BMI      ER10        ; BR IF LOCKED
2259 2CDD AD8B35      LDA      CCBQRM      ; GET REQ MOD
2260 2CE0 C905        CMP      #CRMMAX    ; IN RANGE
2261 2CE2 B0EE        BCS      ERR3A      ; BR IF NOT IN RANGE
2262            ;
2263 2CE4 0A          ASLA
2264 2CE5 AA          TAX
2265 2CE6 BD8A2B      LDA      WVT+1,X    ; GET ROUTINE ADR
2266 2CE9 4B          PHA
2267 2CEA BDB92B      LDA      WVT,X
2268 2CED 4B          PHA
2269 2CEE 60          RTS      ; AND GO TO IT
2270

```


		PAGE		
2271				
2272				
2273				
2274				
2275	2CEF 20DE32	JSR	LOCSEC	; GO GET REQD REL SECTOR
2276				
2277				
2278				
2279	2CF2 20102D	RNXBYT JSR	GETBYT	; GO GET BYTE
2280	2CF5 8D9235	STA	CCBDAT	; PUT IN CCB
2281	2CF8 4C5D33	JMP	GOODIO	; DONE
2282				
2283				
2284				
2285	2CFB 20DE32	RSPBLK JSR	LOCSEC	; GO LOCATE REL SECTOR
2286				
2287				
2288				
2289				
2290	2CFE 20A231	JSR	DTBLN	; GO DECR LEN (NOT RTN IF=0)
2291	2D01 20102D	JSR	GETBYT	; GO GET BYTE
2292	2D04 48	PHA		
2293	2D05 208F31	JSR	MIBDA	; GO MOVE BLOCK ADR AND INCR
2294	2D08 A000	LDY	#0	
2295	2D0A 68	PLA		
2296	2D0B 9142	STA	(ZPGFCB),Y	; SET DATA BYTE
2297	2D0D 4CFE2C	JMP	RNXBLK	; GO FOR NEXT BYTE
2298				
2299				
2300				
2301				
2302	2D10 20A330	JSR	LOCNXB	; LOCATE NEXT BYTE
2303	2D13 B00B	BCS	EOFIN	; BR IF EOF
2304	2D15 B142	LDA	(ZPGFCB),Y	; GET DAT BYTE
2305	2D17 48	PHA		; SAVE IT
2306	2D18 204831	JSR	INCRRB	; INCR REC BYTE
2307	2D1B 208131	JSR	INCSCB	; INCR SEC BYTE
2308	2D1E 68	PLA		; GET SAVED BYTE
2309	2D1F 60	RTS		; RETURN
2310				
2311	2D20 4C4D33	EOFIN JMP	ERROR5	; GO TO EOF RTN
2312				

		PAGE	
2313			
2314			WSPBYT - WRITE SPECIFIC BYTE
2315			
2316		WSPBYT	
2317	2D23 20DE32	JSR	LOCSEC ; GO LOCATE SECTOR
2318			
2319			WNXBYT - WRITE NEXT BYTE
2320			
2321		WNXBYT	
2322	2D26 AD9235	LDA	CCBDAT ; GET THE BYTE
2323	2D29 20422D	JSR	PUTBYT ; GO WRITE BYTE
2324	2D2C 4C5D33	JMP	GOODIO ; DONE
2325			
2326			WSPBLK - WRITE A SPECIFIC BLOCK
2327			
2328		WSPBLK	
2329	2D2F 20DE32	JSR	LOCSEC ; GO LOCATE SECTOR
2330			
2331			WNXBLK - WRITE NEXT BLOCK
2332			
2333		WNXBLK	
2334	2D32 208F31	JSR	MIBDA ; GO MOVE ADR TO ZPG AND DEC
2335	2D35 A000	LDY	#0
2336	2D37 B142	LDA	(ZPGFCB), Y ; GET DATA BYTE
2337	2D39 20422D	JSR	PUTBYT ; GO PUT IT
2338	2D3C 20A231	JSR	DTBLN ; GO DEC BLK LEN (NOT RTN IF = 0)
2339	2D3F 4C322D	JMP	WNXBLK
2340			
2341			PUTBYT - PUT OUT ONE BYTE
2342			
2343		PUTBYT	
2344	2D42 48	PHA	; SAVE DATA BYTE
2345	2D43 20A330	JSR	LOCNXB ; GO LOCATE NEXT BYTE
2346			
2347	2D46 68	PBO	PLA ; GET SAVED BYTE
2348	2D47 9142	STA	(ZPGFCB), Y ; PUT THE BYTE
2349	2D49 A940	LDA	##40 ; SET WRITE SECTOR REQD
2350	2D4B 0DA535	ORA	DCBWRF
2351	2D4E 8DA535	STA	DCBWRF
2352			
2353	2D51 204831	JSR	INCRRB ; INCR REL REC BYTE
2354	2D54 208131	JSR	INCSCB ; INCR SECTOR BYTE
2355	2D57 60	RTS	; DONE
2356			


```

                PAGE
2357            ;
2358            ; FLOCK - LOCK A FILE
2359            ;
2360 2D58 A980    FLOCK LDA    $$B0          ; REMEMBER LOCK
2361 2D5A BD8233 STA    TEMP3
2362 2D5D D005    BNE    LCKGD
2363            ;
2364            ; FUNLCK - UNLOCK A FILE
2365            ;
2366 2D5F A900    FUNLCK LDA    #00          ; REMEMBER UNLOCK
2367 2D61 BD8233 STA    TEMP3
2368            ;
2369            LCKGD
2370            ;
2371 2D64 20EB2B JSR    DOPEN          ; GO OPEN FILE
2372 2D67 AE8033 LDX    TEMP1
2373 2D6A BD9734 LDA    VDFILE+2, X      ; GET FILE USE CODE
2374 2D6D 297F    AND    #$7F          ; TURN OFF LOCK
2375 2D6F 0D8233 ORA    TEMP3
2376 2D72 9D9734 STA    VDFILE+2, X
2377 2D75 203430 JSR    WRVDIR
2378 2D78 4C5D33 JMP    GOODIO
2379            ;
2380            ; FPOSTN - POSITION A FILE
2381 2D7B 20DE32 FPOSTN JSR    LOCSEC      ; GO POSITION
2382 2D7E 4C5D33 JMP    GOODIO          ; DONE
2383            ;
2384

```


PAGE

```

2385 ;
2386 ; FDEL - DELETE A FILE
2387 ;
2388 FDEL
2389 2D81 20EB2B JSR DOPEN ; GO OPEN FILE
2390 ;
2391 2D84 AE8033 FD2 LDX TEMP1 ; SAVED INDEX
2392 2D87 BD9734 LDA VDFILE+2,X ; IS FILE LOCKED
2393 2D8A 1003 BPL FD3 ; BR NOT LOCKED
2394 2D8C 4C5933 JMP ERRR10
2395 ;
2396 FD3
2397 2D8F AE8033 LDX TEMP1 ; GET SAVED INDEX
2398 2D92 BD9534 LDA VDFILE,X ; GET DIR TRACK
2399 2D95 8DA135 STA DCBFDI ; SET AS 1ST FD TRACK
2400 2D98 9DB734 STA VDFILE+34,X ; SAVE IN LC OF FN
2401 2D9B A9FF LDA #$FF ; DELETED FILE MARKER
2402 2D9D 9D9534 STA VDFILE,X ; CLEAR ENTRY
2403 2DA0 BC9634 LDY VDFILE+1,X ; GET DIR SECTOR
2404 2DA3 8CA235 STY DCBFDS ; SET AS 1ST FD SEC
2405 2DA6 203430 JSR WRVDIR ; GO WRITE VOLUME DIR
2406 2DA9 18 CLC
2407 2DAA 205B2F FD4 JSR RFDIR ; GET 1ST FILE DIR SECTOR
2408 2DAD B02A BCS FD7 ; BR IF NO MORE
2409 2DAF 20002F JSR MVFCBD ; MOVE DIR TO ZPG
2410 2DB2 A00C LDY #FDENT ; POINT Y TO 1ST SEC ENT
2411 2DB4 8C8033 FD5 STY TEMP1 ; SAVE Y
2412 2DB7 B142 LDA (ZPGFCB),Y ; GET REACK
2413 2DB9 300B BMI FD6 ; BR IF NONE
2414 2DBB F009 BEQ FD6 ; BR IF END OF FILE
2415 2DBD 48 PHA ; SAVE TRK
2416 2DBE C8 INY
2417 2DBF B142 LDA (ZPGFCB),Y ; GET SECTOR
2418 2DC1 A8 TAY ; TO Y
2419 2DC2 68 PLA ; GET TRK
2420 2DC3 20DF2D JSR FDSUB ; GO FREE SECTOR
2421 2DC6 AC8033 FD6 LDY TEMP1 ; GET DIR INDEX
2422 2DC9 C8 INY ; INCR TO NEXT ENTRY
2423 2DCA C8 INY
2424 2DCB D0E7 BNE FD5 ; BR NOT DONE THIS DIR
2425 2DCD ADA335 LDA DCBCDI ; GET THIS DIR TRK
2426 2DD0 ACA435 LDY DCBCDS ; AND SECTOR
2427 2DD3 20DF2D JSR FDSUB ; AND GO FREE IT
2428 2DD6 38 SEC ; GO
2429 2DD7 B0D1 BCS FD4 ; READ NEXT DIR
2430 FD7
2431 2DD9 20F82F JSR WRVTOC
2432 2DDC 4C5D33 JMP GOODIO
2433 ;
2434 FDSUB
2435 2DDF 38 SEC ; SET FOR RE USE OF SEC
2436 2DE0 20BB32 JSR FRESEC ; GO FREE SECTOR
2437 2DE3 A900 LDA #0 ; CLEAR DCB BIT MAP
2438 2DE5 A203 LDX #3

```


2439	2DE7	9DBC35	FDS1	STA	DCBALS, X
2440	2DEA	CA		DEX	
2441	2DEB	10FA		BPL	FDS1
2442	2DED	60		RTS	
2443					

		PAGE		
2444				
2445			RDIR - PRINT DIRECTORY	
2446				
2447			RDIR	
2448	2DEE 20662C		JSR	DCBSUP
2449	2DF1 20F42F		JSR	RDVTOC
2450	2DF4 A916		LDA	#22 ; SET 21 LINES
2451	2DF6 8D8133		STA	TEMP2
2452	2DF9 20372E		JSR	PRCR ; GO PRINT
2453	2DFC 18		CLC	; FIRST RECORD
2454				
2455	2DFD 200E30	RD1	JSR	RDVDIR ; GO READ REC
2456	2E00 B032		BCS	RD5
2457	2E02 A200		LDX	#0 ; SET INDEX=0
2458	2E04 8E8033	RD2	STX	TEMP1 ; SAVE INDEX
2459	2E07 BD9534		LDA	VDFILE,X ; GET TRACK
2460	2E0A F028		BEQ	RD5 ; BR IF END OF DIR
2461	2E0C 301F		BMI	RD4 ; BR IF DELETED
2462	2E0E E8		INX	
2463	2E0F E8		INX	
2464	2E10 E8		INX	
2465	2E11 8E8233		STX	TEMP3
2466	2E14 BD9534	RD3	LDA	VDFILE,X ; GET CHAR
2467	2E17 20EDFD		JSR	PRINT ; PRINT CHAR
2468	2E1A EE8233		INC	TEMP3
2469	2E1D 201532		JSR	VDINC
2470	2E20 EC8233		CPX	TEMP3
2471	2E23 F005		BEQ	RD3A
2472	2E25 AE8233		LDX	TEMP3
2473	2E28 D0EA		BNE	RD3
2474		RD3A		
2475	2E2A 20372E		JSR	PRCR ; GO PRINT CR
2476	2E2D 201532	RD4	JSR	VDINC ; INCR INDEX
2477	2E30 90D2		BCC	RD2 ; BR IF MORE IN DIR
2478	2E32 B0C9		BCS	RD1 ; GO READ NEXT DIR SECT
2479				
2480	2E34 4C5D33	RD5	JMP	GOODIO ; DONE
2481				
2482			PRCR	
2483	2E37 A98D		LDA	##8D ; CR
2484	2E39 20EDFD		JSR	PRINT ; PRINTED
2485	2E3C CE8133		DEC	TEMP2 ; DEC LINE COUNTER
2486	2E3F D00B		BNE	PRCR1 ; BR IF NOT ZERO
2487	2E41 200CFD		JSR	GETKEY ; WAIT FOR INPUT
2488	2E44 A915		LDA	#21 ; RESET LINE COUNTER
2489	2E46 8D8133		STA	TEMP2
2490	2E49 60	PRCR1	RTS	; DONE
2491				

PAGE

```

2492      ;
2493      ; CLCFB - GET FCB VIA INDEX AND MOVE IT
2494      ;
2495      CLCFB
2496      ;
2497 2E4A 20FC2E      JSR      MVFCBP      ; MOVE FCB PTR TO ZPG
2498 2E4D A000        LDY      #0
2499 2E4F B142        LDA      (ZPGFCB),Y    ; MOVE FCB TO
2500 2E51 99A035      STA      FCB,Y        ; FCB WORK AREA
2501 2E54 C8          INY
2502 2E55 C02A        CPY      #FCBLEN
2503 2E57 D0F6        BNE      CF3
2504      ;
2505 2E59 18          CLC
2506 2E5A 60          RTS      ; DONE
2507      ;
2508      ; RTNFCB - MOVE FCB FROM WORK AREA TO FCB
2509      ;
2510      RTNFCB
2511 2E5B 20FC2E      JSR      MVFCBP      ; MOVE FCB ADR TO ZPG
2512      ;
2513 2E5E A000        LDY      #0
2514 2E60 B9A035      LDA      FCB,Y
2515 2E63 9142        STA      (ZPGFCB),Y
2516 2E65 C8          INY
2517 2E66 C02A        CPY      #FCBLEN
2518 2E68 D0F6        BNE      RF1
2519 2E6A 60          RTS
2520

```


PAGE

```

2521 ;
2522 ; FFMT - EXECUTE FORMAT REQUEST
2523 ;
2524 ; FFMT
2525 2E6B 20662C JSR DCBSUP ; SET UP DCB
2526 2E6E A904 LDA #IBFMT
2527 2E70 205530 JSR DCBIO2
2528 2E73 ADC535 LDA DCBVOL ; SET VOL NO
2529 2E76 49FF EOR #$FF
2530 2E78 BD9033 STA VVOLNO
2531 2E7B A911 LDA #17
2532 2E7D BDBA33 STA VALCA1 ; ALLOCATE BYTE 1
2533 2E80 A901 LDA #1
2534 2E82 BDBB33 STA VALCA2 ; ADD BYTE 2
2535 ;
2536 2E85 A238 LDX #VSECAL-VTOC
2537 2E87 A900 LDA #0
2538 2E89 9D8A33 NT1 STA VTOC,X ; CLEAR SECTOR AREA
2539 2E8C E8 INX
2540 2E8D D0FA BNE NT1
2541 ;
2542 2E8F A20C LDX #3*4 ; START AT TRACK 3
2543 2E91 E08C NT2 CPX #35*4 ; END AT TRACK 35
2544 2E93 F014 BEQ NT4
2545 2E95 A003 LDY #3 ; 4 BYTES OF INFO
2546 2E97 B98433 NT3 LDA ALC10S,Y ; 10 SECTORS ALLOCATE
2547 2E9A 9DC233 STA VSECAL,X
2548 2E9D E8 INX
2549 2E9E 88 DEY
2550 2E9F 10F6 BPL NT3
2551 2EA1 E044 CPX #17*4 ; AT TRACK 17
2552 2EA3 D0EC BNE NT2 ; BR IF NOT
2553 2EA5 A248 LDX #18*4 ; SKIP TO 18
2554 2EA7 D0E8 BNE NT2
2555 ;
2556 2EA9 20F82F NT4 JSR WRVTOC ; WRITE NEW VTOC
2557 ;
2558 2EAC A200 LDX #0
2559 2EAE 8A TXA
2560 2EAF 9D8A34 NT5 STA VOLDIR,X ; CLEAR VOLDIR
2561 2EB2 E8 INX
2562 2EB3 D0FA BNE NT5
2563 ;
2564 2EB5 204230 JSR MVVDBA ; MOVE BUF PTRS
2565 ;
2566 2EB8 A902 LDA #IBCWTS ; WRITE TRACK SECTOR
2567 2EBA 8DF437 STA IBCMD
2568 2EBD A911 LDA #17 ; TRACK 17
2569 2EBF ACBF33 LDY VN0SEC
2570 2EC2 88 DEY
2571 2EC3 88 DEY
2572 2EC4 8DEC37 STA IBTRK ; INTO IOB
2573 2EC7 8D8B34 NT6 STA VDLTRK ; INTO LINK
2574 2ECA 8C8C34 NT7 STY VDLSEC

```


2575	2ECD	C8	INY		
2576	2ECE	8CED37	STY	IBSECT	
2577	2ED1	205830	JSR	DCB101	; GO WRITE
2578	2ED4	AC8C34	LDY	VDLSEC	
2579	2ED7	88	DEY		; DECREMENT SECTOR
2580	2ED8	3005	BMI	NT8	; BR LAST WRITTEN
2581	2EDA	DOEE	BNE	NT7	; BR NOT LAST
2582	2EDC	98	TYA		; LAST, SET LINK TRK=0
2583	2EDD	FOE8	BEQ	NT6	
2584					
2585					
2586	2EDF	20E82E	JSR	DLDSUP	; GO SET UP FOR DOSLDR
2587	2EE2	205537	JSR	WBOOT	; GO WRITE THE BOOT
2588	2EE5	4C5D33	JMP	GOOD10	; DONE
2589					

; NT8

2590			PAGE	
2591			DLDSUP - SET UP FOR DOSLDR	
2592				
2593		DLDSUP		
2594	2EE8 AD8B35	LDA	CCBBSA	
2595	2EEB 8DF137	STA	IBBUFP+1	; START ADR
2596	2EEE A900	LDA	#0	
2597	2EF0 8DF037	STA	IBBUFP	
2598	2EF3 ADC535	LDA	DCBVOL	; VOL
2599	2EF6 49FF	EOR	##FF	
2600	2EF8 8DEB37	STA	IBVOL	
2601	2EFB 60	RTS		
2602				

		PAGE	
2603			
2604			
2605			
2606	2EFC A200	MVFCBP LDX	#0 ; MOVE FCB ADR
2607	2EFE F006	BEG	MVF1
2608	2F00 A202	MVFCBD LDX	#2 ; MOVE FCB DIR BUFF
2609	2F02 D002	BNE	MVF1
2610	2F04 A204	MVFCBS LDX	#4 ; MOVE FCB SECTOR BUFF
2611			
2612		MVF1	
2613	2F06 BD9635	LDA	CFCBAD, X ; DO THE MOVE
2614	2F09 B542	STA	ZPGFCB
2615	2F0B BD9735	LDA	CFCBAD+1, X
2616	2F0E B543	STA	ZPGFCB+1
2617	2F10 60	RTS	
2618			
2619			
2620			
2621		CLRSEC	
2622	2F11 A900	LDA	#0
2623	2F13 AB	TAY	
2624	2F14 9142	CS1 STA	(ZPGFCB), Y
2625	2F16 CB	INY	
2626	2F17 D0FB	BNE	CS1
2627	2F19 60	RTS	
2628			

		PAGE		
2629				
2630				WRSECT - WRITE CURRENT SECTOR IF REQD
2631				
2632			WRSECT	
2633	2F1A 2CA535	BIT	DCBWRF	; GET WRITE REQD FLAG
2634	2F1D 7001	BVS	WRSGD	; BR IF WRITE SECTOR REQD
2635	2F1F 60	RTS		; RTS
2636				
2637			WRSGD	
2638	2F20 20E12F	JSR	MVSBA	; GO MOVE SECT BUFF ADR
2639				
2640	2F23 A902	LDA	#IBCWTS	; GET COMMAND
2641	2F25 204F30	JSR	DCBID	; GO FILL IN IOB AND DO IO
2642				
2643	2F28 A9BF	LDA	#\$BF	; SET WRITE SECTOR REQD BIT OFF
2644	2F2A 2DA535	AND	DCBWRF	
2645	2F2D 8DA535	STA	DCBWRF	
2646	2F30 60	RTS		; DONE
2647				

PAGE

```

2648      ;
2649      ; WRFDIR - WRITE FILE DIRECTRY IF REQD
2650      ;
2651      ; WRFDIR
2652 2F31 ADA535 LDA      DCBWRF      ; GET WRITE REQD FLAG
2653 2F34 3001 BMI      WRFDGO      ; BR IF WRITE DIR REQD
2654 2F36 60 RTS          ; DONE IF NOT
2655      ;
2656      ; WRFDGO
2657 2F37 20482F JSR      MVFDBA
2658      ;
2659 2F3A A902 LDA      #IBCWTS      ; GET WRITE CMD
2660 2F3C 204F30 JSR      DCBIO      ; GO FILL IN IOB AND DO I/O
2661      ;
2662 2F3F A97F LDA      #$7F          ; TURN WRITE DIR REQD BIT OFF
2663 2F41 2DA535 AND      DCBWRF
2664 2F44 8DA535 STA      DCBWRF
2665 2F47 60 RTS          ; DONE
2666      ;
2667      ; MVFDBA - MOVE FILE DIRECTORY BUFF ASDR TO IOB
2668      ;
2669      ; MVFDBA
2670 2F48 AD9835 LDA      CFCBDR      ; MOVE ADR
2671 2F4B 8DF037 STA      IBBUFP
2672 2F4E AD9935 LDA      CFCBDR+1
2673 2F51 8DF137 STA      IBBUFP+1
2674 2F54 AEA335 LDX      DCBCDT      ; GET TRACK
2675 2F57 ACA435 LDY      DCBCDS      ; GET SECTOR
2676 2F5A 60 RTS
2677

```


		PAGE	
2678			
2679			RDFDIR - READ FILE DIRECTORY
2680			
2681		RDFDIR	
2682	2F5B 08	PHP	; SAVE STATUS
2683	2F5C 20312F	JSR WRFDIR	; GO WRITE CURRENT DIR IF REQD
2684	2F5F 20482F	JSR MVFDBA	; GO MOVE DBUFF ADR TO IOB
2685	2F62 20002F	JSR MVFCBD	; MOVE DBUFF ADR TO ZPG
2686	2F65 28	PLP	; GET SAVED STATUS
2687	2F66 B009	BCS RFDNXT	; BR IF RD NEXT
2688			
2689	2F68 AEA135	LDX DCBFDT	; TRACK
2690	2F6B ACA235	LDY DCBFDS	; SECTOR
2691	2F6E 4CB22F	JMP RFDIO1	; GO READ
2692			
2693		RFDNXT	
2694	2F71 A001	LDY #FDLTRK	; GET LINK TRACK
2695	2F73 B142	LDA (ZPGFCB),Y	
2696	2F75 F008	BEQ RFDNL	; NR NO LINK
2697	2F77 AA	TAX	; PUT TRACK INTO X
2698	2F78 C8	INY	
2699	2F79 B142	LDA (ZPGFCB),Y	; SET LINK SECTOR
2700	2F7B AB	TAY	; PUT SECTOR INTO Y
2701	2F7C 4CB22F	JMP RFDIO1	; GO DO I/O
2702			
2703		RFDNL	
2704	2F7F AD8A35	LDA CCBREQ	; THIS A WRITE
2705	2F82 C904	CMP #CRQWR	
2706	2F84 F002	BEQ RFDNL1	; BR IF WRITE
2707	2F86 38	SEC	; SET EOF
2708	2F87 60	RTS	; RETURN
2709			
2710		RFDNL1	
2711	2F88 202932	JSR GETSEC	; GET A SECTOR
2712	2F8B A002	LDY #FDLSEC	
2713	2F8D 9142	STA (ZPGFCB),Y	; PUT IN LINK
2714	2F8F 48	PHA	; SAVE SECTOR
2715	2F90 88	DEY	
2716	2F91 ADBD35	LDA DCBATH	; GET TRACK
2717	2F94 9142	STA (ZPGFCB),Y	; PUT IN LINK
2718	2F96 48	PHA	; SAVE TRACK
2719	2F97 20372F	JSR WRFDGO	; GO WRITE OLD DIR DEC
2720			
2721	2F9A 20112F	JSR CLRSEC	; CLEAN OUT DIR
2722	2F9D A005	LDY #FDFRS	; SET NEW DIR SEC 1ST REL
2723	2F9F ADAC35	LDA DCBDNF	; FILE SECTOR
2724	2FA2 9142	STA (ZPGFCB),Y	
2725	2FA4 C8	INY	
2726	2FA5 ADAD35	LDA DCBDNF+1	
2727	2FA8 9142	STA (ZPGFCB),Y	
2728			
2729	2FAA 68	PLA	; GET SAVED TRACK
2730	2FAB AA	TAX	; INTO X
2731	2FAC 68	PLA	; GET SAVED SECTOR

2732	2FAD	AB		TAY		; INTO Y
2733	2FAE	A902		LDA	#IBCWTS	; SET WRITE CMD
2734	2FB0	D002		BNE	RFDIO2	; GO DO I/O
2735						
2736	2FB2	A901	RFDIO1	LDA	#IBCRTS	; SET READ CMD
2737	2FB4	8EA335	RFDIO2	STX	DCBCDT	; SET CURR TRACK
2738	2FB7	8CA435		STY	DCBCDS	; SET CURR SECTOR
2739	2FBA	204F30		JSR	DCBIO	; GO I/O
2740						
2741	2FBD	A005	RDFDC	LDY	#FDFRS	; GET POINTER TO FIRST REL SECTOR
2742	2FBF	B142		LDA	(ZPGFCB), Y	; GET FRS
2743	2FC1	8DAA35		STA	DCBDFS	; SET INTO DCB
2744	2FC4	18		CLC		
2745	2FC5	6DA835		ADC	DCBDMS	; ADD MAX SECTORS
2746	2FC8	8DAC35		STA	DCBDNF	; PUT INTO DCB
2747						
2748	2FCB	CB		INY		; DO SAME FOR HI BYTE
2749	2FCC	B142		LDA	(ZPGFCB), Y	
2750	2FCE	8DAB35		STA	DCBDFS+1	
2751	2FD1	6DA935		ADC	DCBDMS+1	
2752	2FD4	8DAD35		STA	DCBDNF+1	
2753						
2754	2FD7	18		CLC		
2755	2FDB	60		RTS		; DONE
2756						

		PAGE	
2757			
2758			
2759			
2760			
2761	2FD9 20E12F	RDSECT	
2762			
2763	2FDC A901	JSR	MVSBA ; GO MOVE SECTOR BUFFER ADR
2764	2FDE 4C4F30	LDA	#IBCRIS
2765		JMP	DCBIO ; GO DO I/O
2766			
2767			
2768			
2769	2FE1 AC9A35	MVSBA	
2770	2FE4 AD9B35	LDY	CFCBSB ; GET SECTOR BUFF ADR
2771	2FE7 BCF037	LDA	CFCBSB+1
2772	2FEA BDF137	STY	IBBUFP ; SET IOB SECTOR
2773	2FED AEA635	STA	IBBUFP+1 ; BUFF PTR
2774	2FF0 ACA735	LDX	DCBTRK ; GET TRACK
2775	2FF3 60	LDY	DCBSEC ; GET SECTOR
2776		RTS	; RTN

		PAGE		
2777				
2778			RDVTOC - READ VTOC	
2779			WRVTOC - WRITE VTOC	
2780				
2781		RDVTOC		
2782	2FF4 A901	LDA	#IBCRTS	; READ
2783	2FF6 D002	BNE	VTIO	
2784		WRVTOC		
2785	2FF8 A902	LDA	#IBCWTS	; WRITE
2786				
2787	2FFA AC892B	LDY	AVTOC	; MOVE BUFF ADR
2788	2FFD 8CF037	STY	IBBUFP	
2789	3000 AC8A2B	LDY	AVTOC+1	
2790	3003 8CF137	STY	IBBUFP+1	
2791				
2792	3006 AEC635	LDX	DCBVTN	; GET TRACK
2793	3009 A000	LDY	#0	
2794	300B 4C4F30	JMP	DCBIO	; GO DO I/O
2795				

		PAGE			
2796					
2797				RDVDIR - READ VOLUME DIRECTOR	
2798					
2799			RDVDIR		
2800	300E 08		PHP		; SAVE STATUS
2801	300F 204230		JSR	MVVDBA	
2802					
2803	3012 28		PLP		; GET STATUS
2804	3013 B008		BCS	RVDA	; BR IF R0 NEXT
2805					
2806	3015 AC8C33	RVDC	LDY	VDIRSC	; GET 1ST SECTOR
2807	3018 AE8B33		LDX	VDIRTK	; GET FIRST TRK
2808	301B D00A		BNE	RVDGO	; GO READ
2809					
2810		RVDA			
2811	301D AE8B34		LDX	VDLTRK	; GET LINK TRACK
2812	3020 D002		BNE	RDVC	; BR IF A LINK
2813	3022 38		SEC		; SET END OF DIR
2814	3023 60		RTS		
2815					
2816	3024 AC8C34	RDVC	LDY	VDLSEC	; GET SECTOR
2817		RVDGO			
2818	3027 8E7B33		STX	CVDTRK	; SET CUR TRACK
2819	302A 8C7C33		STY	CVDSEC	; SET CUR SECTOR
2820	302D A901		LDA	#IBCRTS	; GET CMD
2821	302F 204F30		JSR	DCBIO	; GO DO I/O
2822	3032 18		CLC		
2823	3033 60		RTS		
2824					

		PAGE		
2825				
2826			WRVDIR - WRITE VOLUME DIRECTORY SECTOR	
2827				
2828		WRVDIR		
2829	3034 204230	JSR	MVVDBA	
2830				
2831	3037 AE7B33	LDX	CVDTRK	; CURRENT TRACK
2832	303A AC7C33	LDY	CVDSEC	; CURRENT SECTOR
2833	303D A902	LDA	#IBCWTS	; WRITE COMMAND
2834	303F 4C4F30	JMP	DCBID	; GO DO I/O
2835				
2836			MVVDBA - MOVE VOL DIR BUF ADR TO IOB	
2837				
2838		MVVDBA		
2839	3042 AD8B2B	LDA	AVOLDR	; MOVE ADR
2840	3045 8DF037	STA	IBBUFP	
2841	3048 AD8C2B	LDA	AVOLDR+1	
2842	304B 8DF137	STA	IBBUFP+1	
2843	304E 60	RTS		
2844				

		PAGE		
2845				
2846			DCBIO - DO I/O FOR A DCB	
2847				
2848		DCBIO		
2849	304F 8EEC37	STX	IBTRK	; TRACK
2850	3052 8CED37	STY	IBSECT	; SECTOR
2851		DCBIO2		
2852	3055 8DF437	STA	IBCMD	; COMMAND
2853		DCBIO1		
2854	3058 ADC535	LDA	DCBVOL	; VOL
2855	305B 49FF	EOR	##FF	; UNINVERT VOL BITS
2856	305D 8DEB37	STA	IBVOL	
2857	3060 ADC335	LDA	DCBSLT	; SLOT
2858	3063 8DE937	STA	IBSLOT	
2859	3066 ADC435	LDA	DCBDRV	; DRIVE
2860	3069 8DEA37	STA	IBDRVN	
2861	306C ADB035	LDA	DCBSDL	; LENGTH
2862	306F 8DF237	STA	IBDLEN	
2863	3072 ADB135	LDA	DCBSDL+1	
2864	3075 8DF337	STA	IBDLEN+1	
2865	3078 A901	LDA	#1	; IOB TYPE
2866	307A 8DEB37	STA	IBTYPE	
2867				
2868	307D AC872B	LDY	AIOB	; IOB ADR
2869	3080 AD882B	LDA	AIOB+1	
2870	3083 20003D	JSR	DISKIO	; GO DO I/O
2871				
2872	3086 A9FF	LDA	##FF	; RESET VOL
2873	3088 8DEB37	STA	IBVOL	
2874	308B B001	BCS	BADIO	; BR IF BAD
2875	308D 60	RTS		; RTN IF GOOD
2876				
2877	308E AEF537	BADIO	LDX	IBSTAT
2878	3091 8A		TXA	
2879	3092 4920	EOR	#IBVMME	; WAS IT VOL MISMATCH
2880	3094 D008	BNE	BD1	; BR IF NOT
2881	3096 AEF637	LDX	IBSMOD	; GET VOL
2882	3099 A907	LDA	#CREVMM	; SET VOL MM ERR
2883	309B 4C6533	JMP	ERRORB	; GO RTN
2884	309E A908	BD1	LDA	#CREIOE
2885	30A0 4C6533	JMP	ERRORB	; GO RTN
2886				

PAGE

```

2887 ;
2888 ; LOCNXB - LOCATE NEXT BYTE
2889 ;
2890 LOCNXB
2891 30A3 ADB235 LDA DCBCRS ; IS THE CURRENT RELATIVE SECTOR
2892 30A6 CDAE35 CMP DCBCMS ; EQUAL TO THE CURRENT MEM SECTOR
2893 30A9 D008 BNE LNB1 ; BR IF NOT EQ
2894 30AB ADB335 LDA DCBCRS+1
2895 30AE CDAF35 CMP DCBCMS+1
2896 30B1 F066 BEQ LNB8 ; BR IF REQD SECTOR IN MEM
2897 ;
2898 ; LNB1 ; NEED A DIFFERENT SECTOR IN MEM
2899 30B3 201A2F JSR WRSECT ; GO WRITE SECTOR (IF REQD)
2900 ;
2901 30B6 ADB335 LNB2 LDA DCBCRS+1 ; IS CURRENT REL SECTORY
2902 30B9 CDAB35 CMP DCBDFS+1 ; IN CURRENT DIRECTORY (LOW LIMIT)
2903 30BC 901C BCC LNB4 ; BR IF IN A PREVIOUS DIR
2904 30BE D008 BNE LNB3 ; BR IF MAYBE IN THIS ONE
2905 30C0 ADB235 LDA DCBCRS ; TEST LOW BYTES
2906 30C3 CDAA35 CMP DCBDFS
2907 30C6 9012 BCC LNB4 ; BR IF IN PREVIOUS DIR
2908 ;
2909 30C8 ADB335 LNB3 LDA DCBCRS+1 ; IS CURRENT REL SECTOR
2910 30CB CDAD35 CMP DCBDNF+1 ; IN CURRENT DIRECTOR (HI LIMIT)
2911 30CE 9010 BCC LNB6 ; BR IF IN THIS ONE
2912 30D0 D008 BNE LNB4 ; BR IF IN A NEXT DIR
2913 30D2 ADB235 LDA DCBCRS
2914 30D5 CDAC35 CMP DCBDNF
2915 30D8 9006 BCC LNB6 ; BR IF IN THIS ONE
2916 ; ; REQD SECTOR IN A NEXT DIRECTORY
2917 30DA 205B2F LNB4 JSR RDEFDIR ; GO READ NEXT FILE DIR
2918 30DD 90D7 BCC LNB2 ; BR NXT AVAIL
2919 30DF 60 RTS ; RETURN IF EOF DIR
2920 ;
2921 ;
2922 ; LNB6 ; CALCULATE DISPL INTO DIR
2923 30E0 38 SEC
2924 30E1 ADB235 LDA DCBCRS ; REQD REL SECTOR MINUS
2925 30E4 EDAA35 SBC DCBDFS
2926 30E7 0A ASLA ; TIMES 2
2927 30E8 690C ADC #FDENT ; PLUS DISPL TO 1ST
2928 30EA A8 TAY
2929 30EB 20002F JSR MVFCBD ; MOVE DIR ADR TO ZPG
2930 30EE B142 LDA (ZPGFCB),Y ; GET TRACK
2931 30F0 D00F BNE LNB7 ; BR IF NOT ZERO
2932 30F2 AD8A35 LDA CCBREQ
2933 30F5 C904 CMP #CRQWR ; WRITE!
2934 30F7 F002 BEQ LNB7A
2935 30F9 38 SEC
2936 30FA 60 RTS
2937 30FB 202131 LNB7A JSR GNWSEC ; GO GET A NEW SECTOR
2938 30FE 4C0D31 JMP LNBCON
2939 3101 8DA635 LNB7 STA DCBTRK ; SET TRK INTO DCB
2940 3104 C8 INY

```


2941	3105	B142		LDA	(ZPGFCB),Y	; GET SECTOR
2942	3107	8DA735		STA	DCBSEC	; PUT INTO DCB
2943	310A	20D92F		JSR	RDSECT	; GO READ SECTOR
2944	310D	ADB235	LNBCON	LDA	DCBCRS	; MOVE CUR REL SECTOR
2945	3110	8DAE35		STA	DCBCMS	
2946	3113	ADB335		LDA	DCBCRS+1	; TO CUR MEM SECTOR
2947	3116	8DAF35		STA	DCBCMS+1	
2948						
2949			LNBB			
2950	3119	20042F		JSR	MVFCBS	; MOVE SECTOR BUFF ADR TO ZP
2951	311C	ACB435		LDY	DCBCSB	; GET SECT BYTE
2952	311F	18		CLC		; CARRY CLEAR = ALL OK
2953	3120	60		RTS		; DONE
2954						

		PAGE		
2955				
2956				
2957		GNWSEC		
2958	3121	8C8133	STY	TEMP2
2959	3124	202932	JSR	GETSEC
2960	3127	AC8133	LDY	TEMP2
2961	312A	C8	INY	
2962	312B	9142	STA	(ZPGFCB),Y
2963	312D	8DA735	STA	DCBSEC
2964	3130	88	DEY	
2965	3131	ADBD35	LDA	DCBATK
2966	3134	9142	STA	(ZPGFCB),Y
2967	3136	8DA635	STA	DCBTRK
2968				
2969	3139	20042F	JSR	MVFCBS
2970	313C	20112F	JSR	CLRSEC
2971				
2972				
2973	313F	A9C0	LDA	#\$C0
2974	3141	0DA535	DRA	DCBWRF
2975	3144	8DA535	STA	DCBWRF
2976	3147	60	RTS	
2977				

PAGE

```

2978 ;
2979 ; INCRRB - INCREMENT RELATIVE RECORD BYTE
2980 ;
2981 INCRRB
2982 3148 AEB835 LDX DCBCRR ; MOVE BYTE JUST READ OR WRITTEN
2983 314B 8EBC35 STX CCBRRN
2984 314E AEB935 LDX DCBCRR+1
2985 3151 8EBD35 STX CCBRRN+1
2986 3154 AEBA35 LDX DCBCRB ; X=REL BYTE (LOW)
2987 3157 ACBB35 LDY DCBCRB+1 ; Y=REL BYTE HI
2988 315A 8EBE35 STX CCBBYT
2989 315D 8CBF35 STY CCBBYT+1
2990 3160 EB INX ; INC REL BYTE (LOW)
2991 3161 D001 BNE INCR1 ; BR IF NO CARRY
2992 3163 CB INY ; INC REL BYTE (HI)
2993 ;
2994 3164 CCB735 INCR1 CPY DCBRCL+1 ; REL BYTE=REC LENGTH
2995 3167 D011 BNE INCR2 ; BR IF NOT
2996 3169 ECB635 CPX DCBRCL ; TEST LOW BYTES
2997 316C D00C BNE INCR2
2998 316E A200 LDX #0
2999 3170 A000 LDY #0 ; RESET REL BYTE TO ZERO
3000 3172 EEB835 INC DCBCRR ; AND INCR
3001 3175 D003 BNE INCR2 ; RELATIVE RECORD
3002 3177 EEB935 INC DCBCRR+1
3003 ;
3004 317A 8EBA35 INCR2 STX DCBCRB ; SAVE NEW RELATIVE BYTE
3005 317D 8CBB35 STY DCBCRB+1
3006 ;
3007 3180 60 RTS
3008

```


3009			PAGE	
3010			INCSCB - INCREMENT SECTOR BYTE	
3011				
3012		INCSCB		
3013	3181	EEB435	INC	DCBCSB ; INC SECTOR BYTE
3014	3184	D008	BNE	INCS2 ; BR IF NOT FULL
3015	3186	EEB235	INC	DCBCRS ; AND INCR
3016	3189	D003	BNE	INCS2 ; RELATIVE SECTOR
3017	318B	EEB335	INC	DCBCRS+1
3018				
3019				
3020			INCS2	
3021	318E	60	RTS	; DONE
3022				

		PAGE		
3023				
3024				
3025				
3026				
3027	318F AC9235	LDY	CCBBBA	; Y=ADR LOW
3028	3192 AE9335	LDX	CCBBBA+1	; X=ADR HI
3029	3195 8442	STY	ZPGFCB	; PUT ADR INTO ZPG
3030	3197 8643	STX	ZPGFCB+1	
3031				
3032	3199 EE9235	INC	CCBBBA	; INC ADR LOW
3033	319C D003	BNE	MIB1	; BR IF NOT ZERO
3034	319E EE9335	INC	CCBBBA+1	; INC ADR HI
3035	31A1 60	RTS		; DONE
3036				
3037				
3038				
3039				
3040	31A2 AC9035	LDY	CCBBLN	; GET LEN LOW
3041	31A5 D008	BNE	DTB1	; BR IF NOT ZERO
3042	31A7 AE9135	LDX	CCBBLN+1	; GET LEN HI
3043	31AA F007	BEQ	DTB2	; BR IF LEN=0
3044	31AC CE9135	DEC	CCBBLN+1	; DEC LEN (HIGH)
3045	31AF CE9035	DEC	CCBBLN	; DEC LEN (LOW)
3046	31B2 60	RTS		; DONE
3047				
3048	31B3 4C5D33	JMP	GOODIO	; FINISHED BLOCK
3049				

		PAGE		
3050				
3051			FNDFIL - FIND FILE NAME IN VOLUME DIR	
3052				
3053			FNDFIL	
3054	31B6 20F42F	JSR	RDVTOC	; GO GET VTOC
3055	31B9 AD9235	LDA	CCBFN1	; MOVE FN PTR
3056	31BC 8542	STA	ZPGFCB	; TO ZERO PAGE
3057	31BE AD9335	LDA	CCBFN1+1	
3058	31C1 8543	STA	ZPGFCB+1	
3059	31C3 A901	LDA	#1	
3060	31C5 8D8133	FF1	STA	TEMP2
3061	31C8 18		CLC	
3062		FF2		
3063	31C9 200E30	JSR	RDVDIR	; GO GET VDIR SECTOR
3064	31CC B051	BCS	FF4A	
3065	31CE A200	LDX	#0	; SET FOR 1ST FILE
3066				
3067	31D0 8E8033	FF3	STX	TEMP1 ; SAVE INDEX
3068	31D3 BD9534		LDA	VDFILE, X ; GET FILE TRK
3069	31D6 F01F		BEG	FF6 ; BR IF LAST ENTRY
3070	31D8 3022		BMI	FF7 ; BR DELETED ENTRY
3071	31DA A000		LDY	#0 ; X=X+3
3072	31DC E8		INX	
3073	31DD E8		INX	
3074	31DE E8	FF4	INX	
3075	31DF B142		LDA	(ZPGFCB), Y ; GET FN CHAR
3076	31E1 DD9534		CMP	VDFILE, X ; COMPARE TO ENTRY CHAR
3077	31E4 D00A		BNE	FF5 ; BR IF NOT SAME
3078	31E6 C8		INY	
3079	31E7 C020		CPY	#32 ; ALL 32 CHARS
3080	31E9 D0F3		BNE	FF4 ; BR IF NOT
3081	31EB AE8033		LDX	TEMP1 ; GET INDEX
3082	31EE 18		CLC	; FILE FOUND
3083	31EF 60		RTS	; RETURN
3084				
3085		FF5		
3086	31F0 201532		JSR	VDINC
3087	31F3 90DB		BCC	FF3
3088	31F5 B0D2		BCS	FF2
3089				
3090	31F7 AC8133	FF6	LDY	TEMP2 ; LOOKING FOR DELETED
3091	31FA D0C9		BNE	FF1 ; BR IF NOT (DO)
3092				
3093	31FC AC8133	FF7	LDY	TEMP2 ; LOOKING FOR EMPTY
3094	31FF D0EF		BNE	FF5 ; BR IF NOT
3095				
3096		MVFN		
3097	3201 A000		LDY	#0 ; HAVE NEW ENTRY
3098	3203 E8		INX	
3099	3204 E8		INX	
3100	3205 E8	FF8	INX	
3101	3206 B142		LDA	(ZPGFCB), Y ; MOVE FILE NAME
3102	3208 9D9534		STA	VDFILE, X
3103	320B C8		INY	

3104	320C	C020		CPY	#32	
3105	320E	D0F5		BNE	FF8	
3106						
3107	3210	AE8033		LDX	TEMP1	; GET INDEX
3108	3213	38		SEC		; SET NOT OLD
3109	3214	60		RTS		; DONE
3110			VDINC			
3111	3215	18		CLC		
3112	3216	AD8033		LDA	TEMP1	
3113	3219	6923		ADC	#35	
3114	321B	AA		TAX		
3115	321C	E0F5		CPX	#VDFLEN	
3116	321E	60		RTS		
3117			FF4A			
3118	321F	A900		LDA	#0	
3119	3221	AC8133		LDY	TEMP2	
3120	3224	D09F		BNE	FF1	
3121	3226	4C5533		JMP	ERROR9	
3122						

PAGE

3123					
3124				GETSEC - GET A SECTOR	
3125					
3126			GETSEC		
3127	3229	ADBD35	LDA	DCBATK	; GET ALLOCATED TRK
3128	322C	F019	BEQ	GSS1	; BR IF NONE
3129					
3130			GSO		
3131	322E	CEBC35	DEC	DCBALS	; DECREMENT SECTOR NO
3132	3231	300F	BMI	CS2	; BR IF NO SECTORS REM
3133					
3134	3233	18	CLC		
3135	3234	A204	LDX	#4	; 4 BYTE SHIFT
3136	3236	3EBD35	GSS1	ROL	DCBABM-1, X
3137	3239	CA	DEX		; SHIFT BYTE LEFT
3138	323A	D0FA	BNE	GSS1	
3139	323C	90F0	BCC	GSO	; BR IF NO SECTOR
3140					
3141	323E	ADBC35	LDA	DCBALS	; GET ALLOCATED SECTOR
3142	3241	60	RTS		; RETURN
3143					
3144	3242	A900	CS2	LDA	#0
3145	3244	8DBD35	STA	DCBATK	; TRK
3146					
3147	3247	A900	GSS1	LDA	#0
3148	3249	8D8233	STA	TEMP3	; SET SEARCH STATE=0
3149	324C	20F42F	JSR	RDVTOC	; GET VTOC
3150					
3151			GSS2		
3152	324F	18	CLC		
3153	3250	ADBA33	LDA	VALCA1	; GET LAST ALLOCATED TRK
3154	3253	6DBB33	ADC	VALCA2	; AD (+1) OR (-1)
3155	3256	F009	BEQ	GSS3	; BR IF DECK TO ZERO
3156	3258	CDBE33	CMP	VNDTRK	
3157	325B	9015	BCC	GSS5	; BR IF NOT AT OUTER LIMIT
3158	325D	A9FF	LDA	##FF	; SET (-1)
3159	325F	D00A	BNE	GSS4	
3160	3261	AD8233	GSS3	LDA	TEMP3
3161	3264	D038	BNE	ERR9	; GET SEARCH STATE
3162	3266	A901	LDA	#1	; BR IF NOT ZERO
3163	3268	8D8233	STA	TEMP3	; SET (+1)
3164	326B	8DBB33	GSS4	STA	VALCA2
3165	326E	18	CLC		; SET SEARCH STATE = 1
3166	326F	6D7A33	ADC	GENVTN	; SET NEW (+1) OR (-1)
3167	3272	8DBA33	GSS5	STA	VALCA1
3168	3275	8DBD35	STA	DCBATK	; ADD VTOC TRK NO
3169					; SET NEW LAST ALLOCATED
3170	3278	AB	TAY		; PUT IN DCB
3171	3279	0A	ASLA		; ALLOCATED TRACK
3172	327A	0A	ASLA		; TIME 4
3173	327B	AB	TAY		
3174	327C	A204	LDX	#4	
3175	327E	18	CLC		
3176	327F	B9C533	GSS6	LDA	VSECAL+3, Y
					; MOVE BIT MAP BYTE

3177	3282	9DBD35		STA	DCBABM-1,X	
3178	3285	F006		BEQ	GS7	; BR IF NO BITS ON
3179	3287	38		SEC		; SET HAVE A SECTOR
3180	3288	A900		LDA	#0	; CLEAR VTOC BYTE
3181	328A	99C533		STA	VSECAL+3,Y	
3182	328D	88	GS7	DEY		
3183	328E	CA		DEX		
3184	328F	DOEE		BNE	GS6	; BR IF MORE TO MOVE
3185	3291	90BC		BCC	GS2	
3186	3293	20F82F		JSR	WRVTOC	; GO WRITE VTOC
3187	3296	ADBF33		LDA	VNOSEC	; GET NO SECTORS
3188	3299	8DBC35		STA	DCBALS	; SET IN DCB SECTOR BYTE
3189	329C	D090		BNE	GS0	; GO ALLOCATED SECTOR
3190	329E	4C5533	ERR9	JMP	ERROR9	
3191						

		PAGE		
3192				
3193				
3194				
3195				
3196	32A1	ADBD35	LDA	DCBATK ; GET ALLOCATED TRACK
3197	32A4	D001	BNE	FT1 ; BR IF NONE
3198	32A6	60	RTS	; DONE
3199	32A7	4B	PHA	
3200	32A8	20F42F	JSR	RDVTOC ; GET VTOC
3201	32AB	ACBC35	LDY	DCBALS ; GET SECTOS
3202	32AE	68	PLA	; GET TRACK
3203	32AF	18	CLC	; SET FREE
3204	32B0	20BB32	JSR	FRESEC ; GO FREE
3205	32B3	A900	LDA	#0 ; CLEAR ALLOCATED TRK
3206	32B5	BDBD35	STA	DCBATK
3207	32B8	4CF82F	JMP	WRVTOC ; WRITE VTOC
3208				
3209				
3210				
3211				
3212				
3213	32BB	A2FC	FS1	LDX #252 ; 4 BYTE SHIFT
3214	32BD	7EC234	FS2	ROR DCBABM-252, X ; SHIFT IN CARRY
3215	32C0	E8	INX	; NEXT BYTE
3216	32C1	D0FA	BNE	FS2 ; BR IF NOT DONE
3217	32C3	C8	INX	; INC SECTOR NO
3218	32C4	CCBF33	CPY	VNOSEC ; NORMAL
3219	32C7	D0F2	BNE	FS1 ; BR IF NOT
3220				
3221	32C9	0A	ASLA	; TRACK*4
3222	32CA	0A	ASLA	
3223	32CB	A8	TAY	
3224	32CC	F00F	BEQ	FS4
3225	32CE	A204	LDX	#4
3226	32D0	BDBD35	FS3	LDA DCBABM-1, X ; GET BIT MAP BYTE
3227	32D3	19C533	ORA	VSECAL+3, Y ; OR WITH VTOC BM
3228	32D6	99C533	STA	VSECAL+3, Y
3229	32D9	88	DEY	
3230	32DA	CA	DEX	
3231	32DB	D0F3	BNE	FS3
3232	32DD	60	FS4	RTS ; DONE
3233				

PAGE

```

3234 ;
3235 ; LOCSEC - LOCATE SECTOR FOR RECORD I/O
3236 ;
3237 ; RELSEC = (REL REC * RECLen + RELBYTE)/256
3238 ; SECBYT = REMAINDER
3239 ;
3240 LOCSEC
3241 32DE AD8C35 LDA CCBRRN ; RELATIVE RECORD NUMBER
3242 32E1 8DB435 STA DCBCSB ; TO CSB FOR MULT
3243 32E4 8DB835 STA DCBCRR ; AND CRR FOR SAVE
3244 32E7 AD8D35 LDA CCBRRN+1
3245 32EA 8DB235 STA DCBCRS
3246 32ED 8DB935 STA DCBCRR+1
3247 32F0 A900 LDA #0
3248 32F2 8DB335 STA DCBCRS+1 ; HIGH CRS=0
3249 32F5 A010 LDY #16 ; 16 BIT MULT
3250 ;
3251 32F7 AA LS1 TAX ; SAVE MS BYTE
3252 32F8 ADB435 LDA DCBCSB
3253 32FB 4A LSRA ; IF NO CARRY THEN NO PART PROD
3254 32FC B003 BCS LS1A
3255 32FE 8A TXA
3256 32FF 900E BCC LS2
3257 3301 1B LS1A CLC
3258 3302 ADB335 LDA DCBCRS+1 ; FORM PARTIAL PROD
3259 3305 6DB635 ADC DCBRCL
3260 3308 8DB335 STA DCBCRS+1
3261 330B 8A TXA
3262 330C 6DB735 ADC DCBRCL+1
3263 ;
3264 330F 6A LS2 RORA ; MULT BY 2
3265 3310 6EB335 ROR DCBCRS+1
3266 3313 6EB235 ROR DCBCRS
3267 3316 6EB435 ROR DCBCSB
3268 3319 8B DEY ; DEC BIT COUNT
3269 331A D0DB BNE LS1 ; BR IF MORE BITS
3270 ;
3271 331C AD8E35 LDA CCBBYT ; ADD REL BYTE RESULT
3272 331F 8DBA35 STA DCBCRB ; (SAVE REL BYTE)
3273 3322 6DB435 ADC DCBCSB
3274 3325 8DB435 STA DCBCSB
3275 3328 AD8F35 LDA CCBBYT+1
3276 332B 8DBB35 STA DCBCRB+1 ; (SAVE REL BYTE)
3277 332E 6DB235 ADC DCBCRS
3278 3331 8DB235 STA DCBCRS
3279 3334 A900 LDA #0
3280 3336 6DB335 ADC DCBCRS+1
3281 3339 8DB335 STA DCBCRS+1
3282 333C 60 RTS
3283

```


		PAGE			
3284	333D	A901	ERROR1	LDA	#CREFUN
3285	333F	D022		BNE	ERRORA
3286	3341	A902	ERROR2	LDA	#CRERR
3287	3343	D01E		BNE	ERRORA
3288	3345	A903	ERROR3	LDA	#CREMRE
3289	3347	D01A		BNE	ERRORA
3290	3349	A904	ERROR4	LDA	#CREFOP
3291	334B	D016		BNE	ERRORA
3292	334D	A905	ERROR5	LDA	#CREEOF
3293	334F	D012		BNE	ERRORA
3294	3351	A906	ERROR6	LDA	#CREFNF
3295	3353	D00E		BNE	ERRORA
3296	3355	A909	ERROR9	LDA	#CRENSA
3297	3357	D00A		BNE	ERRORA
3298	3359	A90A	ERRR10	LDA	#CREFLK
3299	335B	D006		BNE	ERRORA
3300	335D	A900	GOOD10	LDA	#0 ; STA=0
3301	335F	AA		TAX	
3302	3360	18		CLC	; CARRY=CLR
3303	3361	9003		BCC	RETURN ; GO RETURN
3304	3363	A200	ERRORA	LDX	#0 ; SM=0
3305	3365	38	ERRORB	SEC	; CARRY=SET
3306			RETURN		
3307	3366	08		PHP	
3308	3367	8D9435		STA	CCBSTA ; SET STA
3309	336A	8E9535		STX	CCBSM ; AND SM
3310	336D	205B2E		JSR	RTNFCB ; GO RTN FCB
3311	3370	28		PLP	; GET STATUS
3312	3371	AE7F33		LDX	ENTSTK ; GET ENT STACK
3313	3374	9A		TXS	; RESTORE STACK
3314	3375	60		RTS	; DONE
3315			EC2		
3316					

PAGE

3317

3318

3319

3320

MISC SYSGEN CELLS FOR THIS DOS

3321 3376 01

GENDRN DB 1

; DOS REL NO.

3322 3377 01

GENSRN DB 1

; SYSGEN REL NO.

3323 3378 01

GENDTP DB 1

; DOS TYPE NO.

3324 3379 03

GENTYP DB 3

; SYSGEN TYPE CODE

3325 337A 11

GENVTN DB 17

; VTOC TRACK NO.

3326

		PAGE			
3327					
3328					
3329				MISC DOS WORK CELLS	
3330	337B 00	CVDTRK	DB	0	; CUR VOL DIR TRK
3331	337C 00	CVDSEC	DB	0	; CUR VOL DIR SECTOR
3332	337D 00	CURCCB	DB	0,0	; CURRENT CCB ADR
	337E 00				
3333	337F 00	ENTSTK	DB	0	; ENTRY STACK POINTER
3334	3380 00	TEMP1	DB	0	; TEMP BYTE1
3335	3381 00	TEMP2	DB	0	; TEMP BYTE 2
3336	3382 00	TEMP3	DB	0	; TEMP BYTE 3
3337	3383 00	ENTSLT	DB	0	; BOOT SLOT SAVED
3338	3384 00	ALC10S	DB	0,0,\$F8,\$FF	; ALLOCATATION TRACK BIT MAP
	3385 00				
	3386 F8				
	3387 FF				
3339	3388 0008	BUFADR	DB	@@\$800	; NIBBKLE BUFFER ADR
3340					

PAGE

3341

;

3342

;

VTOC RECORD AREA

3343

;

3344

VTOC

3345 338A 01

VDOST

DB

1

; BOS TYPE

3346 338B 11

VDIRTK

DB

17

; COLUME DIRECTORY SECTOR

3347 338C 0C

VDIRSC

DB

12

; VOLUME DIRECTORY SECTOR

3348 338D 01

VDOSRN

DB

1

; DOS RELEASE NUMBER

3349 338E 01

VGENRN

DB

1

; SYSGEN RELEASE NUMBER

3350 338F 01

VGENTC

DB

1

; SYSGEN TYPE CODE

3351 3390 00

VVOLNO

DB

0

; VOLUME NUMBER

3352 3391

VVOLNM

RMB

32

; VOLUME NAME

3353 33B1 7A

VTDMS

DB

122

; MAX SECTORS IN A FILE DIR

3354 33B2

VSPARE

RMB

8

; SPARES

3355

;

3356 33BA 11

VALCA1

DB

17

; ALOCATION ALGORITHM BYTE 1

3357 33BB 01

VALCA2

DB

1

; AA BYTE2

3358 33BC 00

VALCA3

DB

0

; AA BYTE3

3359 33BD 00

VALCA4

DB

0

; AA BYTE4

3360 33BE 23

VNOTRK

DB

35

; NO TRACKS ON VOL

3361 33BF 0D

VNOSEC

DB

13

; NO SECTORS PER TRACK

3362 33C0 0001

VSECLN

DB

@256

; NO. BYTES PER SECTOR

3363

;

3364 33C2

VSECAL

EQU

*

; SECTOR ALLOCATION AREA

3365

;

SECTORS ALLOCATED BY BIT MAP

3366

;

4 BYTES OF BITS PER TRACK

3367

;

LEFT MOST BIT REPRESENTS SECTOR N

3368

;

WHERE N=NO SECTORS PER TRACK

3369

;

3370

;

3371

3372	33C2		PAGE		
			ORG	VTDC+256	
3373		:			
3374		:			
3375		:		VOLUME DIRECTORY AREA	
3376		:			
3377	348A	VOLDIR			
3378	348B	VDTCDE	RMB	1	; VOLUME DIRECTORY TYPE CODE
3379	348C	VDLTRK	RMB	1	; VD LINK TRACK
3380	348D	VDLSEC	RMB	1	; VD LINK SECTOR
3381	348E	VDNF	RMB	1	; VD NUMBER FILES THIS SECTOR
3382		VDSPAR	RMB	7	; SPARES
3383	3495	VDFILE	EQU	*	; FILE ALLOCATION AREA (7 FILES)
3384		:			EACH FILE:
3385		:			FILE DIR TRK
3386		:			FILE DIR SECTOR
3387		:			FILE USE CODE
3388		:			FILE NAME (32)
3389	3495		ORG	VOLDIR+256	
3390	358A	VDEND	EQU	*	
3391	0100	VDLEN	EQU	*-VOLDIR	
3392	00F5	VDLEN	EQU	*-VDFILE	
3393		:			
3394		:			

PAGE				
3395				
3396				COMMAND CONTROL BLOCK (CCB)
3397				
3398				CCB
3399	358A	CCBREQ	RMB	1 ; USER REQUEST BYTE
3400	0000	CRQNUL	EQU	0 ; 0-NO REQUEST
3401	0001	CRQOPN	EQU	1 ; 1-OPEN FILE
3402	0002	CRQCLS	EQU	2 ; 2-CLOSE FILE
3403	0003	CRQRD	EQU	3 ; 3-READ DATA
3404	0004	CRQWR	EQU	4 ; WRITE DATA
3405	0005	CRQDEL	EQU	5 ; 5-DELETE FILE
3406	0006	CRQDIR	EQU	6 ; 6-READ DIRECTORY
3407	0007	CRQLCK	EQU	7 ; 7-LOCK FILE
3408	0008	CRQUNL	EQU	8 ; 8-UNLOCK FILE
3409	0009	CRQRNM	EQU	9 ; 9-RENAME
3410	000A	CRQPOS	EQU	10 ; 10-POSITION FILE
3411	000B	CRQFMT	EQU	11 ; 11-FORMAT
3412	000C	CRQMAX	EQU	12
3413				
3414		CCBBSA		; FORMAT - BOOT START ADR PAGE
3415	358B	CCBRQM	RMB	1 ; RREQUEST MODIFIER BYTE
3416	0000	CRMNUL	EQU	0 ; NO MODIFIER
3417	0001	CRMNBT	EQU	1 ; R/W - 1 - NEXT BYTE
3418	0002	CRMNBL	EQU	2 ; R/W - 2 - NEXT BLOCK
3419	0003	CRMSBT	EQU	3 ; R/W - 3 - SPECIFC BYTE
3420	0004	CRMSBL	EQU	4 ; R/W - 4 - SPECIFIC BLOCK
3421	0005	CRMMAX	EQU	5
3422				
3423		CCBRRN		; I/O - RELATIVE RECORD NUMBER
3424		CCBFN2		; RENAME - FILE NAME 2 PTR
3425	358C	CCBRLN	RMB	2 ; OPEN - RECORD LENGTH
3426				
3427		CCBBYT		; I/O - RELATIVE BYTE NO (2 BYTES)
3428	358E	CCBVOL	RMB	1 ; OPEN - VOL NO.
3429	358F	CCBDRV	RMB	1 ; OPEN - DRIVE
3430				
3431		CCBBLN		; I/O - BLOCK LENGTH (2 BYTES)
3432	3590	CCBSLT	RMB	1 ; OPEN - SLOT NO
3433	3591	CCBFUC	RMB	1 ; OPEN - FILE USE CODE
3434				
3435		CCBFN1		; OPEN, DELETE, LOCK, UNLOCK, RENAME - FILENAME P
3436		CCBBBA		; BLOCKK I/O - BLOCK BUFFER PTR
3437	3592	CCBDAT	RMB	2 ; BYTE I/O - DATA BYTE
3438				
3439	3594	CCBSTA	RMB	1 ; RESULT STATUS
3440	0001	CREFUN	EQU	1 ; FCB UNALLOCATED
3441	0002	CRERR	EQU	2 ; CCB REQ RANGE ERR
3442	0003	CREMRE	EQU	3 ; REQ MOD RANGE ERR
3443	0004	CREFOP	EQU	4 ; FCB HAS OPEN FILE ERR
3444	0005	CREEOF	EQU	5 ; END OF FILE ON READ
3445	0006	CREFNF	EQU	6 ; FILE NOT FOUND
3446	0007	CREVMM	EQU	7 ; VOL MIS MATCH
3447	0008	CREIOE	EQU	8 ; I/O ERR
3448	0009	CRENSA	EQU	9 ; NO SECTORS AVAILABLE

3449	000A	CREFLK	EQU	10	; FILE LOCKED
3450					
3451	3595	CCBSM	RMB	1	; STATUS MODIFIER
3452	3596	CCBFCB	RMB	2	; FCB PTR
3453	3598	CCBDBP	RMB	2	; DIR BUF PTR
3454	359A	CCBSBP	RMB	2	; SECTOR BUF PTR
3455	359C	CCBSPR	RMB	4	; SPARE
3456	0016	CCBLEN	EQU	*-CCB	; CCB LENGTH
3457	3596	CFCBAD	EQU	CCBFCB	
3458	3598	CFCBDR	EQU	CCBDBP	
3459	359A	CFCBSB	EQU	CCBSBP	
3460					

		PAGE		
3461				
3462				FILE CONTROL BLOCK (FCB) DEFINITION
3463				DCB - FILE DATA CONTROL BLOCK
3464				
3465			FCB	
3466				
3467	35A0	FCBSTA	RMB 1	; FCB STATUS
3468	0000	FCBUNA	EQU \$00	; FCB UNALLOCATED
3469	0080	FCBALC	EQU \$80	; FCB ALLOCATED
3470	0040	FCBFOP	EQU \$40	; FCB HAS OPEN FILE
3471				
3472				DATA CONTROL BLOCK
3473				
3474			FCBDCB	
3475	35A1	DCBFDT	RMB 1	; 1ST FILE DIRECTORY TRACK
3476	35A2	DCBFDS	RMB 1	; 1ST FILE DIRECTORY SECTOR
3477	35A3	DCBCDT	RMB 1	; CURRENT FILE DIRECTORY TRACK
3478	35A4	DCBCDS	RMB 1	; CURRENT FILE DIRECTORY SECTOR
3479	35A5	DCBWRF	RMB 1	; WRITE REQD FLAG
3480				; \$80=WRITE FILE DIR
3481				; \$40=WRITE SECTOR DIR
3482	35A6	DCBTRK	RMB 1	; SECTOR TRACK ADR
3483	35A7	DCBSEC	RMB 1	; SECTOR ADR
3484	35A8	DCBDMS	RMB 2	; MAX NO DIRECTORY SECTORS
3485	35AA	DCBDFS	RMB 2	; CURRENT DIR 1ST REL SECTOR
3486	35AC	DCBDNF	RMB 2	; REL SECTOR OF NXT DIR
3487	35AE	DCBCMS	RMB 2	; SECTOR CURRENTLY IN MEMORY
3488	35B0	DCBSDL	RMB 2	; SECTOR DATA LENGTH
3489	35B2	DCBCRS	RMB 2	; CURRENT RELATIVE SECTOR
3490	35B4	DCBCSB	RMB 2	; CURRENT SECTOR BYTE
3491	35B6	DCBRCL	RMB 2	; RECORD LENGTH
3492	35B8	DCBCRR	RMB 2	; CURRENT RELATIVE REC
3493	35BA	DCBCRB	RMB 2	; CURRENT RELATIVE BYTE
3494				
3495	35BC	DCBALS	RMB 1	; ALLOCATION SECTOR BYTE
3496	35BD	DCBATK	RMB 1	; ALLOCATION TRACK
3497	35BE	DCBABM	RMB 4	; ALLOCATION TRACK SECTOR BIT MAP
3498				
3499	35C2	DCBFUC	RMB 1	; FILE USE CODE
3500	35C3	DCBSLT	RMB 1	; SLOT NUMBER
3501	35C4	DCBDRV	RMB 1	; DRIVE NUMBER
3502	35C5	DCBVOL	RMB 1	; VOLUME DRIVER
3503	35C6	DCBVTN	RMB 1	; VTOC TRACK NUMBER
3504				
3505	35C7	DCBSPR	RMB 3	; SPARES
3506				
3507	0029	DCBLEN	EQU *-FCBDCB	; DCB LENGTH
3508	002A	FCBLEN	EQU *-FCB	; FCB LENGTH
3509				

			PAGE
3510			
3511			DOSLDR - DOS LOADER AND WRITTER
3512			
3513			BOUND 256
3514		DOSLDR	
3515			GARBAGED BOOT REC 0 HERE
3516	3600		RMB 254
3517	36FE 00	GRSPG	DB 0
3518	36FF 00	GRPGC	DB 0
3519			

PAGE

```

3520          SC3
3521          ;
3522          ; READ DOS AFTER BOOT
3523          ;
3524 3700 BEE937      STX      IB SLOT      ; SET BOOT SLOT
3525 3703 BEF737      STX      IBPSLT      ; SET PREVIOUS SLOT
3526 3706 A001        LDY      #1          ; SET PREV DRIVE
3527 3708 BDF837      STA STY      IBPDRV      ;
3528          ; STY IB DRV
3529 3708 ADE037      LDA      NDPGS      ; COPY NO PAGES TO GET
3530 370E BDE137      STA      BRWCNT
3531 3711 A900        LDA      #0
3532 3713 BDEC37      STA      IBTRK      ; SET TRACK 0
3533          ;
3534 3716 ADE237      LDA      BSDSEC      ; COPY START DOS SECTOR
3535 3719 BDED37      STA      IBSECT
3536          ;
3537 371C ADE337      LDA      BGNDOS      ; COPY STARTR DOS ADR
3538 371F BDF137      STA      IBBUFP+1
3539          ;
3540 3722 A901        LDA      #IBCRTS      ; SET READ
3541 3724 BDF437      STA      IBCMD
3542          ;
3543 3727 8A          TXA
3544 3728 4A          LSRA
3545 3729 4A          LSRA
3546 372A 4A          LSRA
3547 372B 4A          LSRA
3548 372C AA          TAX
3549 372D A900        LDA      #0
3550 372F 9DF804      STA      $4FB,X
3551 3732 9D7804      STA      $478,X
3552 3735 209F37      JSR      BOOTIO      ; GO READ DOS
3553          ;
3554          ; DOSINT - INITIALIZE DOS
3555          ;
3556          DOSINT
3557 3738 A2FF        LDX      #$FF
3558 373A 9A          TXS
3559 373B 8EEB37      STX      IBVOL
3560 373E 2093FE      JSR      SETVID
3561 3741 2089FE      JSR      SETKBD
3562 3744 ADF737      LDA      IBPSLT
3563 3747 BD8333      STA      ENTSLT
3564 374A 4A          LSRA
3565 374B 4A          LSRA
3566 374C 4A          LSRA
3567 374D 4A          LSRA
3568 374E BD302B      STA      CS          ; SET ENTRY CURRENT SLOP
3569          ;
3570 3751 4C001E      DI3      JMP      DOSREL      ; GO TO POST INIT ROUTINE
3571 3754 AE EA      DB $A0      ; DUMMY LDR IMM
3572

```

delete

	WBOOT	PAGE			
3573					
3574	3755	ADF137	LDA	IBBUF+1	; GET START OF DOS
3575	3758	8DE337	STA	BGNDOS	; SAVE IT
3576	375B	3B	SEC		
3577	375C	ADE737	LDA	ADOSLD+1	; CALCULATE
3578	375F	EDE337	SBC	BGNDOS	
3579	3762	8DE037	STA	NDPGS	; NO DOS PAGES
3580	3765	8DE237	STA	BSDSEC	
3581					
3582	3768	A900	LDA	#0	
3583	376A	8DEC37	STA	IBTRK	; TRACK=0
3584	376D	8DED37	STA	IBSECT	; SECTOR=0
3585	3770	8DF037	STA	IBBUF	
3586					
3587	3773	ADE737	LDA	ADOSLD+1	; GET BOOT START ADR
3588	3776	8DF137	STA	IBBUF+1	; TO BUF
3589	3779	8DFE36	STA	GRSPG	; TO GARBAGE RECORD
3590					
3591	377C	A90A	LDA	#10	; NO OF BOOT PAGES
3592	377E	8DE137	STA	BRWCNT	; TO BOOT I/O COUNTER
3593	3781	0A	ASLA		; AND
3594	3782	0A	ASLA		; TO
3595	3783	0A	ASLA		; GARBAGE RECORD
3596	3784	8DFF36	STA	GRPGC	
3597					
3598	3787	A902	LDA	#IBCWTS	; SET WRITE
3599	3789	8DF437	STA	IBCMD	
3600					
3601	378C	209F37	JSR	BOOTIO	; GO WRITE BOOT SECTORS
3602					
3603	378F	ADE337	LDA	BGNDOS	; SET START OF DOS
3604	3792	8DF137	STA	IBBUF+1	
3605					
3606	3795	ADE037	LDA	NDPGS	
3607	3798	8DE137	STA	BRWCNT	
3608	379B	209F37	JSR	BOOTIO	; GO WRITE DOS
3609					
3610	379E	60	RTS		; DONE
3611					

A948
EA- LDA #48
NOP

OK

u

none

0

		PAGE		
3612		BOOTIO		
3613	379F ADE537	LDA	BAIOB+1	
3614	37A2 ACE437	LDY	BAIOB	
3615	37A5 20003D	JSR	DISKIO	
3616				
3617	37A8 ACED37	LDY	IBSECT	; GET SECTOR
3618	37AB C8	INY		; INCREMENT TO NEXT
3619	37AC C00D	CPY	#13	; AT END OF TRACK
3620	37AE D005	BNE	BIO1	; BR IF NOT
3621	37B0 A000	LDY	#0	; SET TO SECTOR ZERO
3622	37B2 EEE37	INC	IBTRK	
3623	37B5 BCED37	STY	IBSECT	; SET NEXT SECTOR
3624				
3625	37B8 EEF137	INC	IBBUFP+1	; INCREMENT BUFFER POINTER
3626	37BB CEE137	DEC	BRWCNT	; DECREMENT PAGE COUNTER
3627	37BE D0DF	BNE	BOOTIO	; BR IF NOT DONE
3628	37C0 60	RTS		
3629				
3630				

PAGE

3631				
3632			DOS PATCH AREA 1	
3633				
3634	37C1	DP1	EQU	*
3635			BOUND 256	
3636	3800		ORG	*-\$20
3637		EC3		
3638	37E0 00	NDPGS	DB	0
3639	37E1 00	BRWCNT	DB	0
3640	37E2 00	BSDSEC	DB	0
3641	37E3 00	BGNDOS	DB	0
3642	37E4 E837	BAIOB	DB	@@IOB
3643	37E6 0036	ADOSLD	DB	@@DOSLDR
3644				

PAGE

```

3645      ;
3646      ;      IOB -   INPUT / OUTPUT CONTROL BLOCK
3647      ;      THE IOB IS USED FOR THE INTERFACE
3648      ;      BETWEEN DOS AND THE DISK I/O ROUTINES
3649      ;
3650      IOB
3651 37E8 01      IBTYPE  DB      1      ; IOB TYPE CODE
3652 37E9 07      IBSLOT  DB      7      ; CONTROLLER SLOT NO.
3653 37EA 00      IBDRVN  DB      0      ; DRIVE NUMBER
3654 37EB FF      IBVOL   DB     $FF     ; VOLUME NUMBER
3655 37EC 00      IBTRK   DB      0      ; TRACK NUMBER
3656 37ED 00      IBSECT  DB      0      ; SECTOR NUMBER
3657 37EE FB37    IBDCTP  DB     @@DCT   ;
3658 37F0 0000    IBBUFP  DB     @@0     ; POINTER TO BUFFER
3659 37F2 0001    IBDLEN  DB     @@256   ; DATA LENGTH
3660 37F4 00      IBCMD   DB      0      ; COMMAND
3661      0000      IBCNUL  EQU      0      ; 0-NULL COMMAND
3662      0001      IBCRTS  EQU      1      ; 1-READ TRACK, SECTOR
3663      0002      IBCWTS  EQU      2      ; 2-WRITE TRACK, SECTOR
3664      0004      IBFMT   EQU      4      ; 4-FORMAT DISK
3665      0008      IBBOOT  EQU      8      ; 8-WRITE BOOT
3666 37F5 00      IBSTAT  DB      0      ; STATUS
3667      0080      IBRERR  EQU     $80     ; READ ERR
3668      0040      IBDERR  EQU     $40     ; DRIVE ERR
3669      0020      IBVMME  EQU     $20     ; VOLUME MISMATCH
3670      0010      IBWPER  EQU     $10     ; WRITE PROTECT ERROR
3671 37F6 00      IBSMOD  DB      0      ; STATUS MODIFIER BYTE
3672 37F7 00      IBPSLT  DB      0      ; PREVIOUS SLOT
3673 37F8 00      IBPDRV  DB      0      ; PREVIOUS DRIVE
3674      37F9      IBSPAR  RMB      2      ; IOB SPARES
3675 37FB 00      DCT     DB      0, 1, $EF, $DB
      37FC 01
      37FD EF
      37FE DB

```

3676

PAGE

3677

3678

3679

3680

37FF

ORG

0

3681

FILDIR

3682

0000

FDUCDE

RMB

1

; FILE USE CODE

3683

0001

FDLTRK

RMB

1

; LINK TO NEXT DIR TRACK

3684

0002

FDLSEC

RMB

1

; LINK TO NEXT DIR SECTOR

3685

0003

FDNSA

RMB

1

; NO SECTORS ALLOCATED

3686

0004

FDLSDL

RMB

1

; LAST SECTOR DATA LENGTH

3687

0005

FDFRS

RMB

2

; 1ST RELATIVE SECTOR IN THIS DIR

3688

0007

FDSPAR

RMB

5

; SPARES

3689

3690

000C

FDENT

RMB

1

; START OF FILE ENTRIES (122)

3691

0000

FDTRK

EQU

0

; TRACK

3692

0001

FDSEC

EQU

1

; SECTOR

3693

3694

0100

FDLAST

EQU

FILDIR+256

3695

PAGE
END3696 0000
ASECT PTRS

TSECT PTRS

BSECT PTRS

SYMBOL MAP

A	0001	A	ADOSLD	37E6	A	ADRTAB	1F0C	A
AEC1	3A8F	A	AEC2	3FFF	A	AEND	2B8D	A*
AIOB	2B87	A	AITSTL	E000	A	ALC10S	3384	A
AP1	24AE	A	AS1VT	2062	A	AS1VTL	000A	A
AS2VT	206C	A	AS2VTL	000A	A	ASBRK1	D865	A
ASBRK2	1067	A	ASC1	3800	A	ASC2	3D00	A
ASEOP	00AF	A	ASEOP2	0069	A	ASHM1	0073	A
ASHM2	006F	A	ASIBSW	2B80	A	ASLMEM	0067	A*
ASRUN1	D7D2	A	ASRUN2	0FD4	A	ASSOP	0067	A
ASTART	200A	A	ATSTV	004C	A*	AVOLDR	2B8B	A
AVTOC	2B89	A	B	0002	A	BADIO	308E	A
BAIOB	37E4	A	BD1	309E	A	BFT1	28F8	A
BFT2	2940	A	BFTIB	2959	A	BGNDOS	37E3	A
BIO1	37B5	A	BLDFTB	28E7	A	BOOTIO	379F	A
BOOTSL	002E	A*	BREAK	2052	A	BRWCNT	37E1	A
BSDSEC	37E2	A	BUFADR	3388	A*	CA	2B38	A
CB	2B36	A*	CCB	358A	A	CCBADR	200C	A
CCBBBA	3592	A	CCBBLN	3590	A	CCBBSA	358B	A
CCBBYT	358E	A	CCBDAT	3592	A	CCBDBP	3598	A
CCBDRV	358F	A	CCBFCB	3596	A	CCBFN1	3592	A
CCBFN2	358C	A	CCBFUC	3591	A	CCBLDR	2113	A*
CCBLEN	0016	A	CCBREQ	358A	A	CCBRLN	358C	A
CCBRQM	358B	A	CCBRRN	358C	A	CCBSBP	359A	A
CCBSLT	3590	A	CCBSM	3595	A	CCBSPR	359C	A*
CCBSTA	3594	A	CCBVOL	358E	A	CCHAR	2B7C	A
CD	2B2E	A	CDETAB	1F35	A	CERTN	21E2	A
CF3	2E4F	A	CFCBAD	3596	A	CFCBDR	3598	A
CFCBSB	359A	A	CFTABA	2B18	A	CHAIN	204E	A
CHIN1	2131	A	CHIN2	213C	A	CHRIN	2121	A
CHROUT	2147	A	CINA	2002	A	CIO	0080	A
CL	2B32	A	CLO	252B	A	CL1	2530	A
CL2	2540	A	CLALL	2526	A	CLC1	23DA	A
CLCFCB	2E4A	A	CLOSE	250C	A	CLRCCB	23D6	A
CLRFNS	22C5	A	CLRSEC	2F11	A	CLRSTS	2871	A
CLX	2516	A	CMDETB	201C	A	CMDGO	23A2	A
CMDNO	2B28	A	CMDNTB	298F	A	CMDRTN	21EF	A
CMDSTB	2A0E	A	CMDVT	2B8F	A	CNF	224E	A
CNF1	2259	A	CNFTBS	2B20	A	CNUM	0044	A
CONT	2056	A	COS0	2168	A	COS01	2178	A
COS1	218A	A	COS1A	218D	A	COS2	219B	A
COS3	21A7	A	COS3A	21B7	A	COS4	21BD	A
COS4A	21C6	A	COS5	21CC	A	COS6	21D8	A
COUTA	2004	A	CR	2B34	A	CREEOF	0005	A
CREFLK	000A	A	CREFNF	0006	A	CREFOP	0004	A

CREFUN	0001	A	CREIDE	0008	A	CREMRE	0003	A
CRENSA	0009	A	CRERR	0002	A	CREVMM	0007	A
CRMMA	0005	A	CRMNBL	0002	A	CRMNBT	0001	A
CRMNUL	0000	A*	CRMSBL	0004	A*	CRMSBT	0003	A*
CRQCLS	0002	A	CRQDEL	0005	A	CRQDIR	0006	A
CRQFMT	0008	A	CRQLCK	0007	A	CRQMAX	000C	A
CRQNUL	0000	A*	CRQOPN	0001	A	CRQPOS	000A	A
CRGRD	0003	A	CRQRNM	0009	A	CRQUNL	0008	A
CRQWR	0004	A	CS	2B30	A	CS1	2F14	A
CS2	3242	A	CSERR	2264	A	CURCCB	337D	A*
CUROPT	2B2C	A	CV	2B2C	A	CVDSEC	337C	A
CVDTRK	337B	A	D	0020	A	DBINIT	2076	A
DBRST	20A1	A	DBVECT	2107	A	DCBABM	35BE	A
DCBALS	35BC	A	DCBATK	35BD	A	DCBCDS	35A4	A
DCBCDT	35A3	A	DCBCMS	35AE	A	DCBCRB	35BA	A
DCBCRR	35B8	A	DCBCRS	35B2	A	DCBCSB	35B4	A
DCBDFS	35AA	A	DCBDMS	35AB	A	DCBDNF	35AC	A
DCBDRV	35C4	A	DCBFDS	35A2	A	DCBFDT	35A1	A
DCBFUC	35C2	A	DCBIO	304F	A	DCBIO1	3058	A
DCBIO2	3055	A	DCBLEN	0029	A	DCBRCL	35B6	A
DCBSDL	35B0	A	DCBSEC	35A7	A	DCBSLT	35C3	A
DCBSPR	35C7	A*	DCBSUP	2C66	A	DCBTRK	35A6	A
DCBVOL	35C5	A	DCBVTN	35C6	A	DCBWRF	35A5	A
DCT	37FB	A	DELTA	1F53	A	DENRTS	2BE1	A*
DEPAGE	0040	A	DFNFTB	2B21	A*	DFNFTS	2B7B	A
DG1	27D2	A	DG3	27DF	A	DI3	3751	A
DISKIO	3D00	A	DLDSUP	2EEB	A	DOPEN	2BEB	A
DOSENT	2BC9	A	DOSGO	27CC	A	DOSINT	3738	A*
DOSLDR	3600	A	DOSLNG	1F52	A	DOSREL	1E00	A
DP1	37C1	A*	DPGCNT	1F54	A	DRO	1E16	A
DR1	1E20	A	DR10	1EF5	A	DR11	1F09	A
DR2	1E2F	A	DR3	1E4B	A	DR4	1E58	A
DR5	1E66	A	DR6	1E6C	A	DR7	1E86	A
DR8	1E95	A	DR9	1EB4	A	DRTNI	21F7	A
DRTNO	21F3	A	DSPAGE	0020	A	DTB1	31AF	A
DTB2	31B3	A	DTBLN	31A2	A	EAPND	24AB	A
EAS	2745	A	EAS0	2752	A	EAS1	2757	A
EAS2	2760	A*	EASL1	261C	A*	EAT1	204E	A
EAT2	2BC9	A	EBLD	256D	A	EBLD1	2583	A
EBSV	2541	A	EBSV1	254D	A	EC1	298F	A
EC2	3376	A	EC3	37E0	A	ECAT	2737	A
ECHAIN	26A8	A	ECL1	250B	A	ECLOSE	24F9	A
ECMD	23AE	A	EDEL	247B	A	EDOS	4000	A
EEXEC	26F2	A	EFTABA	2B7E	A	EIBL	262B	A
EIBSV	25AD	A	EIN	2434	A	EINIT	2768	A
EINX	2444	A	ELGO	248C	A	ELOAD	2601	A
ELOCK	2486	A	EM1	000B	A	EM10	0047	A
EM11	0052	A	EM12	005C	A	EM13	006F	A
EM14	0080	A	EM2	000B	A	EM3	000B	A
EM4	000B	A	EM5	000E	A	EM6	0019	A
EM7	0027	A	EMB	0036	A	EM9	003E	A
EMAXF	2465	A	EMDTB	2B08	A	EML	0091	A
EMON	2447	A	EMPR	280E	A	EMPR1	2812	A
EMSG	2A70	A	ENFA	27E6	A	ENM1	2459	A
ENOMON	2451	A	ENTSLT	3383	A	ENTSTK	337F	A

ED1	24CA	A	ED3	24D6	A	ED4	24D9	A*
ED5	24E0	A	ED6	24E6	A*	EOFIN	2D20	A
EOPEN	24B6	A	EPOS	2706	A	EPR	2421	A
EPRX	2431	A	ER10	2CD5	A	EREAD	26BE	A
EREN	2493	A	ERNU1	27EE	A	ERR2	2BE2	A
ERR3A	2CD2	A	ERR9	329E	A	ERROR	27F0	A
ERROR1	333D	A*	ERROR2	3341	A	ERROR3	3345	A
ERROR4	3349	A*	ERROR5	334D	A	ERROR6	3351	A*
ERROR9	3355	A	ERRORA	3363	A	ERRORB	3365	A
ERRR10	3359	A	ERUN	2692	A	ESAVE	258F	A
ESTATE	2B7D	A	ESYNTAX	27E2	A	EUNLK	248A	A
EWRITE	26B3	A	EX0	271B	A	EX1	272B	A
EX1A	2728	A	EX2	2736	A	EXP1	2711	A
EXP2	2714	A	F01	2C69	A	F01A	2C8C	A
F02	2C00	A	F03	2C3E	A	FASB	2B81	A
FASBL	0006	A	FCB	35A0	A	FCBALC	0080	A*
FCBDCB	35A1	A	FCBFOP	0040	A*	FCBLEN	002A	A
FCBSTA	35A0	A*	FCBUNA	0000	A*	FCLOSE	2C96	A
FD2	2D84	A*	FD3	2D8F	A	FD4	2DAA	A
FD5	2DB4	A	FD6	2DC6	A	FD7	2DD9	A
FDEL	2DB1	A	FDENT	000C	A	FDFRS	0005	A
FDLAST	0100	A*	FDLSL	0004	A*	FDLSEC	0002	A
FDLTRK	0001	A	FDNSA	0003	A*	FDS1	2DE7	A
FDSEC	0001	A*	FDSPAR	0007	A*	FDSUB	2DDF	A
FDTRK	0000	A*	FDUCDE	0000	A*	FF1	31C5	A
FF2	31C9	A	FF3	31D0	A	FF4	31DE	A
FF4A	321F	A	FF5	31F0	A	FF6	31F7	A
FF7	31FC	A	FF8	3205	A	FFMT	2E6B	A
FILDIR	0000	A	FILSRC	287A	A	FLOCK	2D58	A
FLS1	2884	A	FLS1A	2889	A	FLS2	2898	A
FLS3	289A	A	FLS4	28A6	A	FN1	0020	A
FN1ADR	2006	A	FN2	0010	A	FN2ADR	2008	A
FNAME1	2B3B	A	FNAME2	2B5B	A	FNDFIL	31B6	A
FOPEN	2BE5	A	FPOSTN	2D7B	A	FREAD	2CC0	A
FRESEC	32BB	A	FRETRK	32A1	A	FRNME	2CA2	A
FS1	32BB	A	FS2	32BD	A	FS3	32D0	A
FS4	32DD	A	FT1	32A7	A	FTAB	2000	A
FUNLCK	2D5F	A	FWRITE	2CD8	A	GENDRN	3376	A*
GENDTP	3378	A*	GENSRN	3377	A*	GENTYP	3379	A*
GENVTN	337A	A	GETBYT	2D10	A	GETKEY	FDOC	A
GETNUM	23E1	A	GETSEC	3229	A	GN2	23E7	A
GN3	23F2	A	GN4	2418	A	GN5	241A	A
GNBC	23CC	A	GNWSEC	3121	A	GNXTC	23BB	A
GNXTCR	23CB	A	GO	2054	A	GOODIO	335D	A
GRPGC	36FF	A	GRSPG	36FE	A	GS0	322E	A
GS1	3236	A	GS2	324F	A	GS3	3261	A
GS4	326B	A	GS5	3272	A	GS6	327F	A
GS7	328D	A	GSS1	3247	A	HOME	FC58	A*
IAS1	208E	A	IAS1A	2095	A	IAS2A	20D4	A
IBASVT	204E	A	IBBOOT	000B	A*	IBBRK	E3E3	A
IBBUFP	37F0	A	IBCHN	E836	A	IBCMD	37F4	A
IBCNUL	0000	A*	IBCONT	E003	A	IBCRIS	0001	A
IBCWTS	0002	A	IBDCTP	37EE	A	IBDERR	0040	A*
IBDLEN	37F2	A	IBDRVN	37EA	A	IBFMT	0004	A
IBGO	E000	A	IBHMEM	004C	A	IBLMEM	004A	A

IBPDRV	37F8	A	IBPSLT	37F7	A	IBRERR	0080	A*
IBRUN	269D	A	IBSECT	37ED	A	IBSLOT	37E9	A
IBSMOD	37F6	A	IBSOP	00CA	A	IBSOV	00CC	A
IBSPAR	37F9	A*	IBSTAT	37F5	A	IBTRK	37EC	A
IBTYPE	37E8	A	IBVMME	0020	A	IBVOL	37EB	A
IBVT	2058	A	IBVTL	000A	A	IBWPER	0010	A*
ICFD	278A	A	ICFD1	27A3	A	ICFD2	2794	A*
ICFD3	278C	A	ICFD4	279E	A	IFB	2103	A
IFBL	001D	A	IIB1	2082	A	IMBITS	2B3A	A
INCR1	3164	A	INCR2	317A	A	INCRRB	3148	A
INCS2	318E	A	INCSCB	3181	A	INITA	20A2	A
INITAA	209E	A	INITB	20C6	A	INITC	20C9	A
INITD	20DD	A	INITE	20EB	A	INITF	2103	A
INOPTS	2B2B	A	INPRT	FE8B	A	INSDS2	F88E	A
INSW	0038	A	IOB	37E8	A	IOBLDR	211A	A*
ISTATE	2B1A	A	ITSTV	0020	A	L	0008	A
LBUFD	2B26	A	LBUFF	0200	A	LCKGD	2D64	A
LD2	264F	A	LD3	2680	A	LDREGS	2204	A
LENGTH	002F	A	LNB1	30B3	A	LNB2	30B6	A
LNB3	30C8	A	LNB4	30DA	A	LNB6	30E0	A
LNB7	3101	A	LNB7A	30FB	A	LNB8	3119	A
LNBCON	310D	A	LOCNxB	30A3	A	LOCSEC	32DE	A
LS1	32F7	A	LS1A	3301	A	LS2	330F	A
MC	0040	A	MFERR	27EA	A	MFULL	268C	A
MI	0020	A	MIB1	31A1	A	MIBDA	318F	A
MD	0010	A	MODECK	21F9	A	MONMOD	2B27	A
MSB1	2FE7	A*	MULT	FB63	A*	MVBP1	2866	A
MVBUFP	2864	A	MVF1	2F06	A	MVECBD	2F00	A
MVFCBP	2EFC	A	MVFCBS	2F04	A	MVFDBA	2F48	A
MVFN	3201	A	MVFN1	2859	A	MVFN1A	285B	A
MVISW	2965	A	MVDSW	297A	A	MVSBA	2FE1	A
MVVDBA	3042	A	NDPGS	37E0	A	NEPAGE	1F51	A
NPB	0080	A	NPE	0040	A	NSPAGE	1F50	A
NT1	2E89	A	NT2	2E91	A	NT3	2E97	A
NT4	2EA9	A	NT5	2EAF	A	NT6	2EC7	A
NT7	2ECA	A	NT8	2EDF	A	NUM1	0008	A
NUM2	0004	A	NXTEXC	27AA	A	OCTD	2776	A
OPEN	24B8	A	OPNSUP	2826	A	OPT1L	000A	A
OPTAB1	2A40	A	OPTAB2	2A4A	A	OPTAB3	2A54	A
ORG1	1E00	A	ORG2	2000	A	ORTN	2200	A
ORTN1	2210	A	OS1	2855	A	OSTATE	2B1B	A
OUTPRT	FE95	A	OUTSVT	200E	A	OUTSW	0036	A
PB0	2D46	A*	PRCIFR	221D	A	PRCR	2E37	A
PRCR1	2E49	A	PRCRIF	2213	A	PRINT	FDED	A
PROMPT	0033	A	PRRTN	21DF	A	PUTBYT	2D42	A
R	0004	A	RBYTE	27BB	A	RD1	2DFD	A
RD2	2E04	A	RD3	2E14	A	RD3A	2E2A	A
RD4	2E2D	A	RD5	2E34	A	RDFDC	2FBD	A*
RDFDIR	2F5B	A	RDIR	2DEE	A	RDONE	1E13	A
RDSECT	2FD9	A	RDVC	3024	A	RDVDIR	300E	A
RDVTOC	2FF4	A	REPAGE	1F4F	A	RETURN	3366	A
RF1	2E60	A	RFDIO1	2FB2	A	RFDIO2	2FB4	A
RFDNL	2F7F	A	RFDNL1	2F88	A	RFDNXT	2F71	A
RNXBLK	2CFE	A	RNXBYT	2CF2	A	RSPAGE	1F4E	A
RSPBLK	2CFB	A	RSPBYT	2CEF	A	RTNFCB	2E5B	A

RUN	2050	A	RVDA	301D	A	RVDC	3015	A*
RVDGO	3027	A	RVT	2BA9	A	RWP1	26D4	A
RWP2	26D7	A	RWP2A	26E0	A	RWP3	26E9	A
RWPOSN	26C9	A	RWPR	26F1	A	S	0010	A
SAT1	2000	A	SAT2	2B87	A	SC0	2224	A
SC0A	2233	A	SC1	2076	A	SC1A	2244	A
SC1X	2236	A	SC2	2BC9	A	SC3	3700	A
SCNCMD	221E	A	SERR1	22C2	A	SERR2	239F	A
SETKBD	FE89	A	SETVID	FE93	A	SN1	22CB	A
SN10	22D2	A	SN11	22F8	A	SN2	227A	A
SN3	2281	A	SN4	2286	A	SN5	228E	A
SN6	2293	A	SN7	22A5	A	SN8	22AC	A
SDPTS	22FB	A	SP1	230F	A	SP2	2333	A
SP3	2335	A	SP4	233F	A	SP5	2366	A
SP6	2376	A	SP7	2388	A	SP8	238B	A
START	2000	A	SV1	25C6	A	SV1A	25C9	A
SV2	25D1	A	SV3	25F0	A	SVA	2B25	A
SVINS	2B1E	A	SVDUTS	2B1C	A	SVREGS	215B	A
SVRGS	215E	A	SVSTK	2B22	A	SVX	2B23	A
SVY	2B24	A	SYNTAX	2267	A	TEMP1	3380	A
TEMP1A	2B29	A	TEMP2	3381	A	TEMP2A	2B2A	A
TEMP3	3382	A	TFUCR	28E6	A	TSINIT	28A8	A
TSNXT	28B0	A	TSR	28BF	A	TSST	28BA	A
TSTEXC	28C5	A	TSTFUC	28DA	A	TSTOPN	28C0	A
TXC1	28D8	A	TXC2	28D9	A	V	0040	A
VALCA1	33BA	A	VALCA2	33BB	A	VALCA3	33BC	A*
VALCA4	33BD	A*	VDEND	358A	A*	VDFILE	3495	A
VDFLEN	00F5	A	VDINC	3215	A	VDIRSC	338C	A
VDIRTK	338B	A	VDLEN	0100	A*	VDLSEC	348C	A
VDLTRK	348B	A	VDNF	348D	A*	VDOSRN	338D	A*
VDOST	338A	A*	VDSPAR	348E	A*	VDTCDE	348A	A*
VGENRN	338E	A*	VGENTC	338F	A*	VNOSEC	33BF	A
VNOTRK	33BE	A	VOLDIR	348A	A	VSECAL	33C2	A
VSECLN	33C0	A*	VSPARE	33B2	A*	VTDMS	33B1	A
VTIO	2FFA	A	VTDC	338A	A	VVOLNM	3391	A*
VVOLND	3390	A	WBOOT	3755	A	WNXBLK	2D32	A
WNXBYT	2D26	A	WRFDGO	2F37	A	WRFDIR	2F31	A
WRSECT	2F1A	A	WRSGO	2F20	A	WRVDIR	3034	A
WRVTOC	2FF8	A	WSPBLK	2D2F	A	WSPBYT	2D23	A
WVT	2BB9	A	ZPGFCB	0042	A	ZPGWRK	0040	A

NO ERROR LINES

SOURCE CK. = 0041 OBJ. CK. = 0000

0 LOCAL REGIONS (63 MAX)

REMAINING TABLE SPACE = 1718

IN RANGE FLAG COUNT= 0

DISC SOURCE FILE (HEX)= 0300-0393